



SAS® Tech Report

JUNE 10, 2008

Dear Readers,

We're off to a great start. I've received many e-mails from you with feedback on the May SAS *Tech Report*. That's good news. Many of you found the Usage Note: FASTats: Frequently Asked-For Statistics highly useful, so we've decided to add another category, **RESOURCES**, to the newsletter. The new category will be updated frequently to direct you to hard-to-find information that you can bookmark and save for later.

As always, if you have questions, ideas, tips or suggestions, please drop me a line.

Waynette Tubbs

techeditor@sas.com

Editor, SAS Tech Report

SAS News

A Quick Look at RSS

An FAQ from SAS

There just isn't enough time to search the Internet each day for all of the new information you need to make better business decisions, take fewer financial risks or learn more about your favorite hobby. But, there is a way to get the information you need *and* have it delivered to you automatically every time it's updated. How? With RSS, or really simple syndication. Read the FAQs found at <http://www.sas.com/news/feature/RSS-FAQ.html> to learn the benefits of RSS.

SAS® Global Forum Papers: The Best of the Best

We are continuing our highlight of the top papers presented at the SAS Global Forum. You can read the [award-winning papers online](#) or check out the [complete list of papers](#).

Global Forum Paper Series: Developing a Dashboard to Aid in Effective Project Management

M. Paige Borden, Maureen Murray and Ali Yorkos discuss the vast improvements offered by the development and implementation of the CMS IR Technical Support Dashboard using SAS Enterprise Guide®, SAS/GRAPH® and SAS BI Dashboard 3.1.
<http://www2.sas.com/proceedings/forum2008/052-2008.pdf>

Global Forum Paper Series: A Step-by-Step Introduction to PROC REPORT

David Lewandowski has broken PROC REPORT into a more user-friendly format, which still allows you to create up to 90 percent of the reports as the more difficult-to-use manual.
<http://www2.sas.com/proceedings/forum2008/079-2008.pdf>

SAS® and Sun: Delivering Scalability, Security and Availability Enterprisewide

For current users of SAS and Sun IT infrastructures or those contemplating a future Sun integration, the Webcast found at <http://www.sas.com/events/partners/153469/main.html> offers an overview of the new SAS Enterprise Intelligence Platform powered by Sun's Datacenter of the Future Knowledge Center. After the Webcast, explore the center's collection of reference architectures, white papers and other materials.
<http://www.sas.com/partners/directory/sun/knowledge.html>

Visual Studio 2005 Visual Basic Code Snippets

IntelliSense Code Snippets speed development by allowing you to insert ready-made snippets of code into your projects.

This sample will be periodically updated with new snippets. If you have ideas for snippets, feel free to leave feedback and we'll look into adding them.

ADONet.VB.snippet

Description: Uses an OleDbDataAdapter to retrieve data from SAS using the IOM Data Provider.

```
'Create a SAS Workspace and initialize it.
'Insert a snippet to create a workspace.

'Add the SAS object to the ObjectKeeper. If the object is not
'in the ObjectKeeper, it cannot be used by the IOM Data Provider.
Dim obObjectKeeper As New SASObjectManager.ObjectKeeper
obObjectKeeper.AddObject(1, "MyServer", obSAS)

'Get the requested data using ADO.NET
Dim selectCommandText As String = "$QueryString$"
Dim obAdapter As New
System.Data.OleDb.OleDbDataAdapter(selectCommandText, _
    "provider=sas.iomprovider.1; SAS Workspace ID=" &
obSAS.UniqueIdentifier)

'Copy data from the adapter into the data set
Dim obDS As New DataSet()
obAdapter.Fill(obDS, "sasdata")

obObjectKeeper.RemoveObject(obSAS)
```

ADODB.VB.snippet

Description: Gets data using the IOM Data Provider and ADODB through COM interop.

```
'Create a SAS Workspace and initialize it.
'Insert a snippet to create a workspace.

'Add the SAS object to the ObjectKeeper. If the object is not
'in the ObjectKeeper, it cannot be used by the IOM Data Provider.
Dim obObjectKeeper As New SASObjectManager.ObjectKeeper
obObjectKeeper.AddObject(1, "MyServer", obSAS)

'Get the formatted data using ADODB.
Dim adoConnection As ADODB.Connection = New ADODB.Connection()
```

```
Dim adoRecordset As ADODB.Recordset = New ADODB.Recordset
Dim adoCommand As ADODB.Command = New ADODB.Command
```

```
adoConnection.Open("provider=SAS.IOMProvider.1; SAS Workspace ID=" & _
    obSAS.UniqueIdentifier, "", "", 0)
```

```
adoCommand.ActiveConnection = adoConnection
adoCommand.CommandText = "$QueryString$"
adoCommand.CommandType = ADODB.CommandTypeEnum.adCmdUnknown
adoCommand.Properties("SAS Formats").Value = "_ALL_"
adoRecordset.Open(adoCommand, System.Reflection.Missing.Value, _
    ADODB.CursorTypeEnum.adOpenDynamic, ADODB.LockTypeEnum.adLockReadOnly, 1)
```

```
'Fill an ADO.NET DataSet using the ADODB Recordset
Dim obAdapter As New System.Data.OleDb.OleDbDataAdapter
Dim sasdata As New System.Data.DataSet
obAdapter.Fill(sasdata, adoRecordset, "sasdata")
```

```
obObjectKeeper.RemoveObject(obSAS)
```

CreateOMRConfFile.VB.snippet

Description: Inserts code to call WriteConfigFile, which creates a metadata config file.

```
Dim obObjectFactory As New SASObjectManager.ObjectFactory
Dim serverdef As New SASObjectManager.ServerDef
Dim logindef As New SASObjectManager.LoginDef
```

```
' *****
```

```
' TODO: Configure where files will be saved by setting the
' following boolean statements.
```

```
' *****
```

```
Dim bSaveLoginForAllUsers As Boolean = True
Dim bSaveServerForAllUsers As Boolean = True
```

```
Dim strAllUsersFilePath As String = Environment.GetFolderPath( _
    Environment.SpecialFolder.CommonApplicationData) & _
    "\SAS\MetadataServer\oms_serverinfo.xml"
```

```
Dim strCurrentUserFilePath As String = Environment.GetFolderPath( _
    Environment.SpecialFolder.ApplicationData) & _
    "\SAS\MetadataServer\oms_serverinfo.xml"
```

```
Dim strCurrentUserLoginFilePath As String = Environment.GetFolderPath( _
    Environment.SpecialFolder.ApplicationData) & _
    "\SAS\MetadataServer\oms_userinfo.xml"
```

```

' *****
' Configure the SAS® Metadata Server information.
' TODO: Modify these values for your SAS Metadata Server.
' *****

serverdef.BridgeEncryptionAlgorithm = "SASProprietary"
serverdef.BridgeEncryptionLevel =
SASObjectManager.EncryptionLevels.EncryptUserAndPassword
serverdef.ClassIdentifier = "2887E7D7-4780-11D4-879F-00C04F38F0DB"
serverdef.Name = "SAS Metadata Server"
serverdef.Port = $ServerDef.Port$
serverdef.MachineDNSName = $ServerDef.MachineDNSName$
serverdef.Protocol = SASObjectManager.Protocols.ProtocolBridge
serverdef.DomainName = $ServerDef.Domain$
Dim strRepository As String = $Repository$

```

```

' *****
' Configure the SAS Metadata Server login information.
' TODO: Modify these values for your SAS Metadata Server.
' *****

logindef.DomainName = serverDef.DomainName
logindef.LoginName = $LoginDef.LoginName$
logindef.Name = loginDef.LoginName
logindef.Password = $LoginDef.Password$

```

If (bSaveServerForAllUsers AndAlso bSaveLoginForAllUsers) Then

```

' *****
' Save server and login information for all users.
' *****

obObjectFactory.WriteConfigFile( _
    strAllUsersFilePath, _
    serverdef, _
    logindef, _
    SASObjectManager.RepositoryTypes.RepositoryOMR, _
    strRepository)

```

Elseif (bSaveServerForAllUsers) Then

```

' *****
' Save server information for all users and login
' information for the current user.
' *****

obObjectFactory.WriteConfigFile( _

```

```

        strAllUsersFilePath, _
        serverdef, _
        Nothing, _
        SASObjectManager.RepositoryTypes.RepositoryOMR, _
        strRepository)

    obObjectFactory.WriteConfigFile( _
        strCurrentUserLoginFilePath, _
        Nothing, _
        logindef, _
        SASObjectManager.RepositoryTypes.RepositoryNone, _
        Nothing)
Else
    ' *****
    ' Save server and login information for current users.
    ' *****

    obObjectFactory.WriteConfigFile( _
        strCurrentUserFilePath, _
        serverdef, _
        logindef, _
        SASObjectManager.RepositoryTypes.RepositoryOMR, _
        strRepository)

    ' It's a good idea to clean up conflicting login
    ' information to ensure that your configuration is using
    ' the correct username and password.
    Dim fiCurrentUserLoginFile As System.IO.FileInfo = New _
        System.IO.FileInfo(strCurrentUserLoginFilePath)

    If (fiCurrentUserLoginFile.Exists) Then _
        fiCurrentUserLoginFile.Delete()
End If

```

COM_DCOMConnection.VB.snippet

Description: Creates a COM or DCOM connection to a SAS Workspace Server.

```

Dim obObjectFactory As New SASObjectManager.ObjectFactory
Dim obServerDef as New SASObjectManager.ServerDef
Dim obSAS As SAS.Workspace

```

```

' *****
' Configure the SAS Metadata Server information.
' TODO: Modify these options as appropriate.
' *****

```

```
obServerDef.Protocol = SASObjectManager.Protocols.ProtocolCom
obServerDef.MachineDNSName = $MachineDNSName$
```

```
' *****
```

```
' Create the connection.
```

```
' *****
```

```
obSAS = obObjectFactory.CreateObjectByServer("MyServer", True, obServerDef, "", "")
```

IOMBridgeConnection.VB.snippet

Description: Creates an IOM Bridge connection to a SAS Workspace Server.

```
Dim obObjectFactory As New SASObjectManager.ObjectFactory
Dim obServerDef as New SASObjectManager.ServerDef
Dim obSAS As SAS.Workspace
```

```
*****
```

```
' Configure the SAS Metadata Server information.
```

```
' TODO: Modify these options as appropriate.
```

```
' *****
```

```
obServerDef.Protocol = SASObjectManager.Protocols.ProtocolCom
obServerDef.MachineDNSName = $MachineDNSName$
obServerDef.BridgeEncryptionAlgorithm = "SASProprietary"
obServerDef.BridgeEncryptionLevel =
SASObjectManager.EncryptionLevels.EncryptUserAndPassword
obServerDef.Port = $Port$
obServerDef.Protocol = SASObjectManager.Protocols.ProtocolBridge
```

```
*****
```

```
' Create the connection.
```

```
' *****
```

```
obSAS = obObjectFactory.CreateObjectByServer("MyServer", True, obServerDef, $UserName$,
$Password$)
```

ConnectionByLogicalName.VB.snippet

Description: Creates a connection to a SAS IOM Server, such as the SAS Metadata Server, that is defined in metadata.

```
Dim obObjectFactory As New SASObjectManager.ObjectFactory
Dim obSAS As SAS.Workspace
```

```
' *****
```

```
' Create the connection.
```

```
' This assumes that your metadata configuration files are in the
default
```

```
' location or that the location of your configuration files is stored
```

```
' in the registry. If this is not the case, then you should call...
' obObjectFactory.SetMetadataFile systemFile, userFile, false
' *****
obSAS = Nothing
Set obSAS = obObjectFactory.CreateObjectByLogicalName("MyServer", True,
$LogicalName$, "")
```

CreatePool.VB.snippet

Description: Creates a connection pool of workspaces for use in an application that needs to make frequent use of a workspaces with minimal wait time.

```
Dim obObjectFactory As New SASObjectManager.ObjectFactory
Dim obServerDef as New SASObjectManager.ServerDef
Dim obLoginDef as New SASObjectManager.LoginDef
Dim objectPool As SASObjectManager.ObjectPool = Nothing

Dim strPoolName As String = "$PoolName$"

Try
'Check to be sure that the pool is not already running.
objectPool = obObjectFactory.ObjectPools.Item(strPoolName)

If Not objectPool Is Nothing Then
    Return
End If

Catch ex As Exception
    Trace.Write( _
        "Pool not created yet, creating pool." & _
        "Exception message: " & _
        ex.Message)
End Try

Try

' *****
' Configure the SAS Metadata Server information.
' TODO: Modify these options as appropriate.
' *****
obServerDef.MachineDNSName = $MachineDNSName$
obServerDef.Protocol = SASObjectManager.Protocols.ProtocolBridge
obServerDef.Port = $Port$
obServerDef.BridgeEncryptionAlgorithm = "SASProprietary"
```

```

obServerDef.BridgeEncryptionLevel =
SASObjectManager.EncryptionLevels.EncryptUserAndPassword
obServerDef.ShutdownAfter = 30
obServerDef.RunForever = False
obServerDef.RecycleActivationLimit = 30
obServerDef.MaxPerObjectPool = 5

' *****
' Configure the login information and pooling parameters.
' TODO:  Modify these options as appropriate.
' *****
obLoginDef.LoginName = $UserName$
obLoginDef.Password = $Password$
obLoginDef.MinSize = 2
obLoginDef.MinAvail = 0

' *****
' Create the ObjectPool.
' *****
objectPool =
obObjectFactory.ObjectPools.CreatePoolByServer(strPoolName,
obServerDef, obLoginDef)

Catch ex As Exception
    Trace.Write("Exception while creating object pool:  " & ex.Message)
    Throw
End Try

' *****
' The following code demonstrates how to create
' a pooled object and return the object back to
' the pool when the application is done with it.
.

' Dim obpooledworkspace As New SASObjectManager.PooledObject
' obpooledworkspace = objectPool.GetPooledObject("", "", 10000)
' Dim obSAS As SAS.Workspace
' obSAS = obpooledworkspace.SASObject

' debug.print "Pooled Workspace Running On:  " &
obSAS.Utilities.HostSystem.DNSName

' Return the object to the pool.

```

```
' obSAS = Nothing
' obPooledWorkspace.ReturnToPool()
' obpooledworkspace = Nothing
' *****
```

SASLSEvents.VB.snippet

Description: Adds a method to initialize the SASLanguageEvents and all of the required method signatures for the event delegates.

```
'Enable Events on the LanguageService object.
Public WithEvents obLS As SAS.LanguageService

'*****
'Define LanguageService Event Handler Routines.
'*****
Private Sub obLS_DatastepStart() Handles obLS.DatastepStart
    'Debug.Write("DatastepStart" & vbCrLf)
End Sub

Private Sub obLS_DatastepComplete() Handles obLS.DatastepComplete
    'Debug.Write("DatastepComplete" & vbCrLf)
End Sub

Private Sub obLS_ProcStart(ByVal Procname As String) Handles
obLS.ProcStart
    'Debug.Write("Proc Start:  " + Procname & vbCrLf)
End Sub

Private Sub obLS_ProcComplete(ByVal Procname As String) Handles
obLS.ProcComplete
    'Debug.Write("Proc Complete:  " + Procname & vbCrLf)
End Sub

Private Sub obLS_SubmitComplete(ByVal Sasrc As Integer) Handles
obLS.SubmitComplete
    'Debug.Write("Submit Complete:  " + Sasrc & vbCrLf)
End Sub

Private Sub obLS_StepError() Handles obLS.StepError
    'Debug.Write("Error in SAS" & vbCrLf)
End Sub
```

SASFileService.VB.snippet

Description: Creates a temporary fileref and uses it to download output generated by SAS back to the client as a byte array. This action requires a connection to already be established.

```
' *****  
' Define instances of interfaces used.  
' *****  
Dim obFS As SAS.FileService  
Dim obFref As SAS.Fileref  
Dim outstring As String  
Dim obLS As SAS.LanguageService  
  
' *****  
' Create a temporary fileref to store dynamically  
' generated output.  
' *****  
obFS = $sasWorkspace$.FileService  
outstring = ""  
obFref = obFS.AssignFileref("out", "TEMP", "", "", outstring)  
  
' *****  
' TODO: Replace with your own logic to generate  
' output in SAS that needs to be downloaded.  
' *****  
obLS = $sasWorkspace$.LanguageService  
Dim sascode As String  
  
' Example:  
' sascode = "goptions reset=all gsfname=out gsfmode=replace dev=pdf;  
proc gchart data=sashelp.class;vbar age / discrete;run;quit;"  
  
' *****  
' Submit the sample code to generate output.  
' *****  
obLS.Submit(sascode)  
  
' *****  
' Stream back the output file from SAS.  
' *****  
Dim obStreamHelper As New SCRIPTOLib.StreamHelper  
Dim obBinaryStream As SAS.BinaryStream  
Dim PDFGraph As Byte()  
obBinaryStream =  
obFref.OpenBinaryStream(SAS.StreamOpenMode.StreamOpenModeForReading)
```

```
obStreamHelper = CreateObject("SASScripto.StreamHelper")
PDFGraph = obStreamHelper.ReadBinaryArray(obBinaryStream, 0)
obBinaryStream.Close()
```

```
' *****
' Here is an example of writing the byte array to
' a file you could add to your code.
' *****
'Dim fs As New System.IO.FileStream("c:\sample.pdf",
IO.FileMode.Create, IO.FileAccess.Write)
'fs.Write(PDFGraph, 0, PDFGraph.Length)
```

<http://support.sas.com/kb/26/145.html>

Silently Install Hot Fixes for SAS[®] 9.1.3 Foundation SAS[®], SAS[®] Enterprise Guide[®] and SAS[®] Add-In for Microsoft Office

Follow these steps to silently install hot fixes for Foundation SAS, SAS Enterprise Guide, and SAS Add-In for Microsoft Office:

1. Download the hot fix to any folder (for example, **c:\<hotfix_id>**).
2. Right-click on the hot fix, and select **WinZIP ► Open with WinZIP**.
3. Verify that the **Use folder names** box is checked and then **Extract to c:\<hotfix_id>**.
4. Open a command prompt, and then type **cd c:\<hotfix_id>** and press **Enter**.
5. Submit the following command:

```
setup.exe /r /s
```

When the command is submitted, the hot fix will proceed to install silently.

To verify that the hot fix installed successfully, review the hot fix log:

Foundation SAS hot fixes write a log to
<!SASROOT>\<hotfix_id>wn.log.

SAS Enterprise Guide hot fixes write a log to
**\Program Files\SAS\Enterprise Guide
X\<hotfix_id>wn.log**.

SAS Add-in for Microsoft Office hot fixes write a log to
**\Program Files\SAS\Add-In for Microsoft
Office\<hotfix_id>wn.log**.

<http://support.sas.com/kb/21/020.html>

Place Symbols at ZIP Code Locations on a US Map

This sample uses the Annotate facility to place symbols at ZIP code locations on a map. The SASHELP.ZIPCODE data set containing the ZIP code coordinates can be downloaded from the Maps Online (<http://support.sas.com/maponline>) web site. <http://support.sas.com/kb/31/419.html>

```
/* Create a data set containing ZIP codes. */
data myzip;
  input zip;
  cards;
35004
85003
71601
80002
6001
19701
20001
32007
83201
60001
46001
50001
66002
40003
70001
3901
20601
1001
27513
;
run;

/* Sort the data set by ZIP codes. */
proc sort data=myzip;
  by zip;
run;

/* Create a data set containing the */
/* X and Y values for my ZIP codes. */
data longlat;
  /* In case there are duplicate ZIP codes, rename
  X and Y from the SASHELP.ZIPCODE data set. */
  merge myzip(in=mine)
        sashelp.zipcode(rename=(x=long y=lat)keep=x y zip);
  by zip;
```

```

/* Keep if the ZIP code was in my data set. */
  if mine;
/* Convert longitude, latitude in degrees to radians */
/* to match the values in the map data set. */
  x=atan(1)/45*long;
  y=atan(1)/45*lat;
/* Adjust the hemisphere */
  x=-x;
  /* Keep only the ZIP, X and Y variables */
  keep zip x y;
run;

/* Create an annotate data set to place a symbol at the
  ZIP code locations. */
data anno;
/* Use the X and Y values from the LONGLAT data set. */
  set longlat;
/* Set the data value coordinate system. */
/* Set the function to label. */
/* Set the size of the symbol to .75. */
/* Set a FLAG variable to signal annotate observations. */
  retain xsys ysys '2' function 'label' size .75 flag 1 when 'a';
/* Set the font to the Special font. */
  style='special';
/* The symbol is a star. */
  text='M';
/* Specify the color for the symbol. */
  color='red';
/* Output the observation to place the symbol. */
  output;
run;

/* Combine the map data set with the annotate data set. */
data all;
  /* Subset out the states that you do not want. */
  /* The FIPS code of 2 is Alaska, 15 is Hawaii, */
  /* and 72 is Puerto Rico. */
  set maps.states(where=(state not in(2 15 72))) anno;
run;

/* Project the combined data set. */
proc gproject data=all out=allp;
  id state;
run;

```

```

quit;

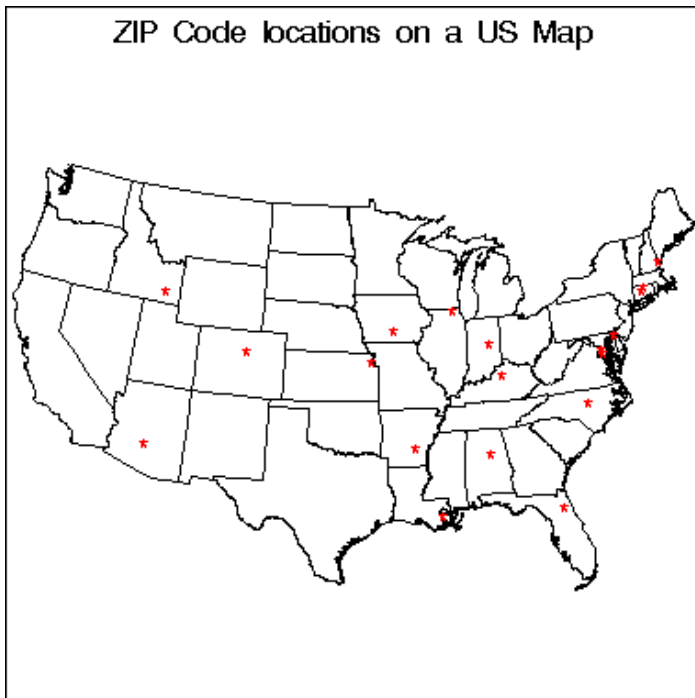
/* Separate the projected data set into a map and an annotate data set.
*/
data map dot;
  set allp;
  /* If the FLAG variable has a value of 1, it is an annotate */
  /* observation; otherwise, it is a map data set observation. */
  if flag=1 then output dot;
  else output map;
run;

/* Define the pattern for the map. */
pattern1 v=me c=black r=50;

/* Define the title for the map. */
title 'ZIP Code locations on a US Map';

/* Generate the map and place the symbols at ZIP code locations. */
proc gmap data=map map=map;
  id state;
  choro state / anno=dot nolegend;
run;
quit;

```



<http://support.sas.com/kb/31/419.html>

Creating a Needle Plot with PROC SGPLOT

This sample uses PROC SGPLOT to produce a needle plot with overlaid plot lines.

```
/* Define a format to order the horizontal axis and
   the format for the values on the vertical axis. */
proc format;
  value matfmt
    1='1 mo'
    2='3 mo'
    3='6 mo'
    6='1 yr'
    7='2 yr'
    8='3 yr'
    9='5 yr'
    11='10 yr '
    14='30 yr';
  picture yldfmt
    low-high='9%';
run;

/* Create sample data. */
data yield;
  input matcur yldcur matlmo yldlmo matlyr yldlyr;
  label yldcur='January 2008' yldlmo = 'December 2007' yldlyr =
'January 2007';
  format matlyr matfmt. matlmo matfmt. matcur matfmt.
    yldcur yldfmt. yldlmo yldfmt. yldlyr yldfmt.;
datalines;
1      2.63    1      2.76    1      4.97
2      2.86    2      3.04    2      5.12
3      2.86    3      3.37    3      5.16
6      2.69    6      3.31    6      5.07
7      2.36    7      3.19    7      4.89
8      2.39    8      3.2     8      4.8
9      2.86    9      3.53    9      4.75
11     4.3     11     4.6     11     4.94
14     4.28    14     4.55    14     4.85
;
run;

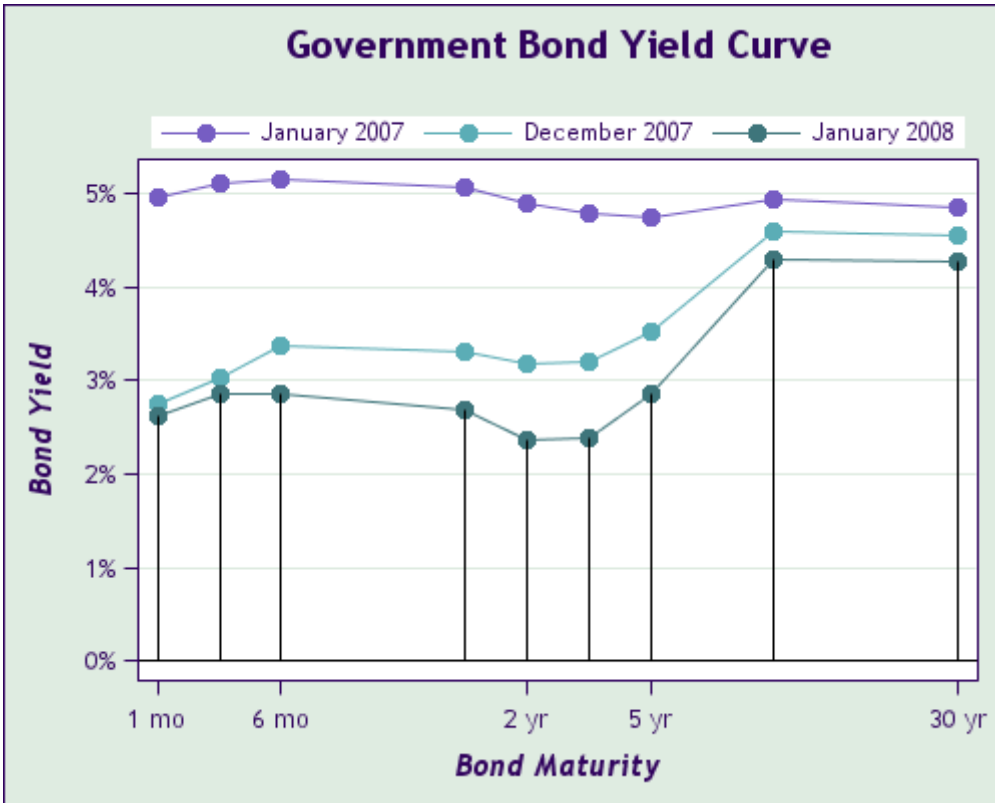
title 'Government Bond Yield Curve';
title2 ' ';
```

```

ods listing style=ocean ;

/* Create the graph with the SGPLOT procedure.  Each SERIES statement
   produces a line and the NEEDLE statement produces the needle in the
   graph. */
proc sgplot data=yield cycleattrs;
  series x=mat1yr y=yld1yr / markers
    markerattrs=(symbol=circlefilled size=10px)
    lineattrs=(pattern=solid) name='n1';
  series x=mat1mo y=yld1mo / markers
    markerattrs=(symbol=circlefilled size=10px)
    lineattrs=(pattern=solid) name='n2';
  series x=matcur y=yldcur / markers
    markerattrs=(symbol=circlefilled size=10px)
    lineattrs=(pattern=solid) name='n3';
  needle x=matcur y=yldcur / lineattrs=graphreference;
  yaxis grid label='Bond Yield';
  xaxis type=linear values=(1 2 3 6 7 8 9 11 14) label='Bond
Maturity';
  keylegend 'n1' 'n2' 'n3' / position=top noborder;
run;
title;
title2;

```



<http://support.sas.com/kb/31/507.html>

Events

Discovery 2008

<http://www.sas.com/events/jmp/>

June 16 – 17
Cary, NC

Certification Exam Fair

<http://support.sas.com/certify/promo.html>

July 1 – 31

M2008

<http://www.sas.com/events/dmconf/index.html>

October 27 – 28
Las Vegas

Resources

FASTats: Frequently Asked-For Statistics

<http://support.sas.com/kb/30/333.html>

SAS[®] 9.2 Highlights and Resource Center

This application is directed to a wide range of map data users: SAS/GRAPH customers, educators, students and general map users.

<http://support.sas.com/software/>

SAS[®] Maps Online

This application is directed to a wide range of map data users: SAS/GRAPH customers, educators, students and general map users.

<http://support.sas.com/rnd/datavisualization/mapsonline/>

Multimedia

2-for-1 Training in July

Take a SAS Training class in July and a co-worker takes the class for **FREE**.

<http://support.sas.com/training/discounts/2for1.html>

New Multimedia e-Lectures Available Now

Learn more about e-Lectures by viewing a FREE e-Lecture, SAS[®] 9.2 Changes and Enhancements for Base SAS[®].

http://www.sas.com/apps/elearning/elearning_category_welcome.jsp

Cody's Data Cleaning Techniques Using SAS[®], Second Edition

Thoroughly updated for SAS[®]9, Ron Cody's second edition develops and demonstrates data cleaning programs and macros addressing tasks that nearly every SAS programmer needs.

<http://www.sas.com/apps/pubscat/bookdetails.jsp?pc=61703>

Webcasts

Outperform the Competition with Analytics

On Demand

<http://www.sas.com/events/cm/165049/index.html>

Forecast Value Added Analysis: Step by Step

On Demand

<http://www.sas.com/events/cm/176129/index.html>