

The value of metadata

Bas van Bakel, OCS Consulting, Rosmalen, The Netherlands

ABSTRACT

The SAS Metadata server is used to store metadata in a repository on a centralised location to support the enterprise intelligence platform. Although this repository contains data that can be of value to an end user, not all metadata is directly accessible.

This paper will demonstrate how this information can be retrieved and presented to an end user. The examples are extracted from a version of the SAS Information Delivery Portal which OCS Consulting has set up to function as a proof of concept environment to illustrate our vision of how web technology can be used to allow an organisation to improve collaboration between people and across departments.

INTRODUCTION

The value that metadata can add to a company is often underestimated. This paper will provide some simple examples of how metadata can be applied to be of added value to a pharmaceutical company. The examples include browsing metadata, setting metadata used by the SAS environment and extracting metadata.

WHAT IS METADATA?

Several definitions of metadata exist. There are some sophisticated definitions like 'Metadata is the data that describes the structure and workings of an organization's use of information, and which describes the systems it uses to manage that information.' or 'Metadata is structured, encoded data that describes characteristics of information-bearing entities to aid in the identification, discovery, assessment, and management of the described entities.', but the most commonly used definition of metadata is 'Metadata is data about data'.

Let's explain this further through an example: Suppose you find a sticky note on your desk, on which somebody has written the value '0.034'. Without more information about this piece of data you would not know what it means and what you are supposed to do with it. The first questions that will probably arise are "What does '0.034' mean?", "Who has written me this note?", "Where does this data come from?". After a second look you see in the corner of the sticky note the name of your colleague and you give him a call. He specifies the value was the result -the p-value- of a specific statistical test. He then explains to you the details of how this value was obtained. The information given by your colleague is metadata. It provides a meaning to the data '0.034'. Even the name written in the corner of the sticky note is metadata as it describes the source of the data.

Metadata can describe:

- the meaning/description of the data;
- the origin of the data;
- where the data resides;
- which applications/persons may and do access the data;
- who is responsible for the correctness of the data;

Please note the definition of metadata, 'metadata is data about data', implies metadata itself is data and can therefore also have metadata.

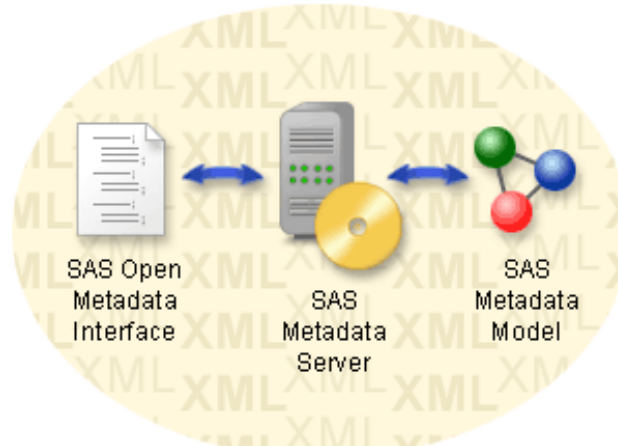
Metadata can provide meaningful and valuable information. The way Google exploits metadata is an example of how metadata can be of use. Google derives statistics about the number of links to a certain webpage or document taken into account the importance of the pages linking to them. These statistics are a good indication about the number of persons who think this page or document is important enough to link to, and therefore the reliability and reputability of that page can be estimated. Metadata can also be very useful when performing an impact analysis of a technical change within an organization. If data about the way a certain derivation is performed is stored in the metadata an overview can easily be generated if the derivation needs to be changed.

PhUSE 2007

METADATA AND SAS

Metadata creation and maintenance can be very time-consuming and therefore expensive. To get the most out of metadata and reduce the maintenance costs it is advisable to have a centralized method of creating, storing, searching and managing metadata. In the document "Getting started with SAS 9.1.3 Open Metadata Interface" the SAS Open Metadata Architecture is described. It is stated that this architecture provides such a centralized method of creating, storing, searching and managing metadata:

- the SAS Metadata Server; a central, shared location for storing metadata;
- the SAS Open Metadata Interface; an application programming interface (API) that provides access to the server from a variety of programming environments, including Java, COM/DCOM, and SAS;
- the SAS Metadata Model; a set of metadata types that are used for saving metadata on the server;
- an XML transport format and XML representation of metadata; which makes it easy to transform the metadata to HTML and other standard XML representations.



Source: "Getting Started with SAS 9.1.3 Open Metadata Interface", SAS Institute

QUERYING AND USING METADATA IN SAS

SAS stores the available metadata in the SAS Metadata Server and provides means to access this metadata with the SAS Open Metadata Interface. To assist the user in managing this metadata, SAS developed the SAS management console, which is a graphical user interface to visually manage the metadata.

This chapter will give some examples about how to query and use metadata in a pharmaceutical environment. In these examples a dataset named DEMOGRAP will be used. The metadata of this dataset is imported in the Data Library Manager through the SAS Management Console. An overview of this dataset is shown below:

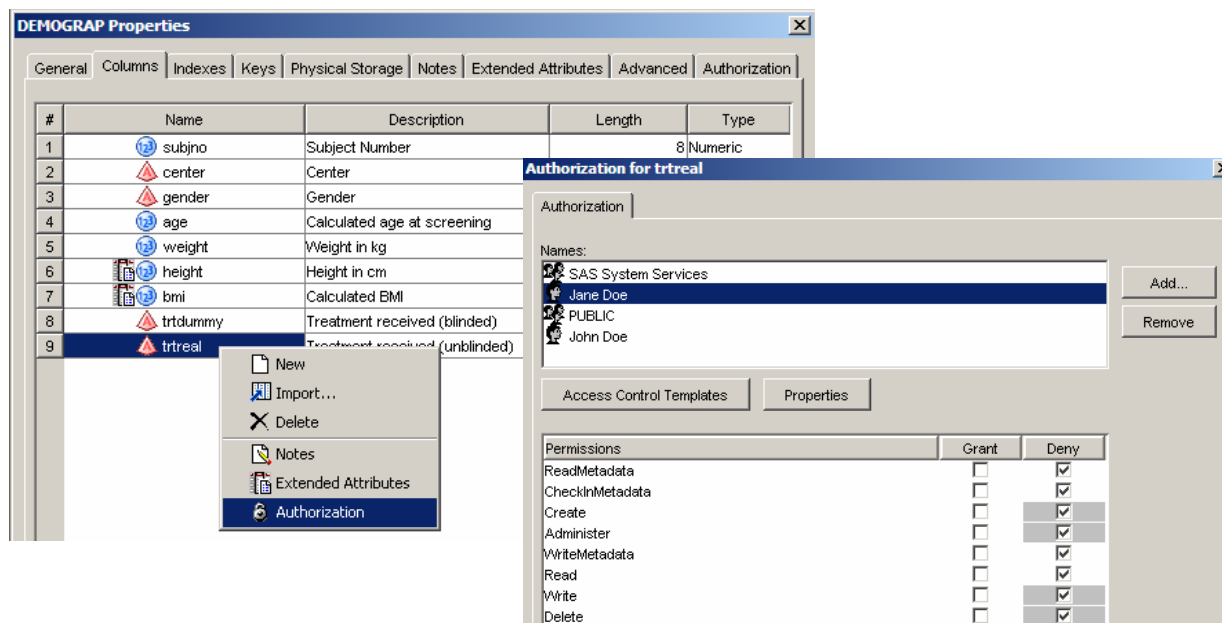
The screenshot displays the SAS Management Console interface. On the left is a tree view of the console's components, including Environment Management, Data Library Manager, and Metadata Manager. The main window shows the 'DEMOGRAP Properties' dialog box, which is currently on the 'Columns' tab. This tab displays a table of the dataset's columns.

#	Name	Description	Length	Type
1	subjno	Subject Number	8	Numeric
2	center	Center	4	Character
3	gender	Gender	1	Character
4	age	Calculated age at screening	8	Numeric
5	weight	Weight in kg	8	Numeric
6	height	Height in cm	8	Numeric
7	bmi	Calculated BMI	8	Numeric
8	trtdummy	Treatment received (blinded)	20	Character
9	trtreal	Treatment received (unblinded)	20	Character

PhUSE 2007

EXAMPLE 1: USING METADATA TO HIDE COLUMNS

When looking at the columns of the DEMOGRAP dataset, please notice columns TRTDUMMY and TRTTREAL. These columns contain for each subject the information about the treatment received. Suppose this study is blinded and the Clinical Trial Statistician named Jane Doe is currently not allowed to see the actual treatment received. Therefore the column TRTTREAL which contains the actual treatment should not be accessible to her and she should use the dummy treatment information available in the column TRTDUMMY. It is possible to hide the column TRTTREAL for her and make it inaccessible. To do this, change the Authorization of this specific column within the SAS Management Console and make sure you 'Deny' the permission for Jane Doe.



As this dataset is only accessible through the Metadata Server, the LIBNAME statement with the META engine has to be used. When accessing the data via the Metadata Server, the authorization specified in the metadata server will be taken into account. Therefore the Unblinded column is not available to Jane Doe.

```
libname phuse meta library='saslib' host='localhost' port=8561 protocol='bridge'
user='janedoe' pw='password' repname='Foundation';
```

	Subject Number	Center	Gender	Calculated age at screening	Weight in kg	Height in cm	Calculated BMI	Treatment received (blinded)
1	101001	NL01	F	32	69	172	23.323418064	A
2	101002	NL01	M	22	71	184	20.971172023	B
3	101003	NL01	M	19	86	184	25.401701323	A
4	101004	NL01	F	29	80	186	23.124060585	B
5	101005	NL01	M	35	64	167	22.948115745	A
6	102001	GB01	M	32	77	178	24.30248706	B
7	102002	GB01	F	41	55	165	20.202020202	A
8	102003	GB01	M	38	85	181	25.94548396	B
9	102004	GB01	M	28	81	173	27.064051589	A
10	102005	GB01	M	41	75	186	21.678806798	B

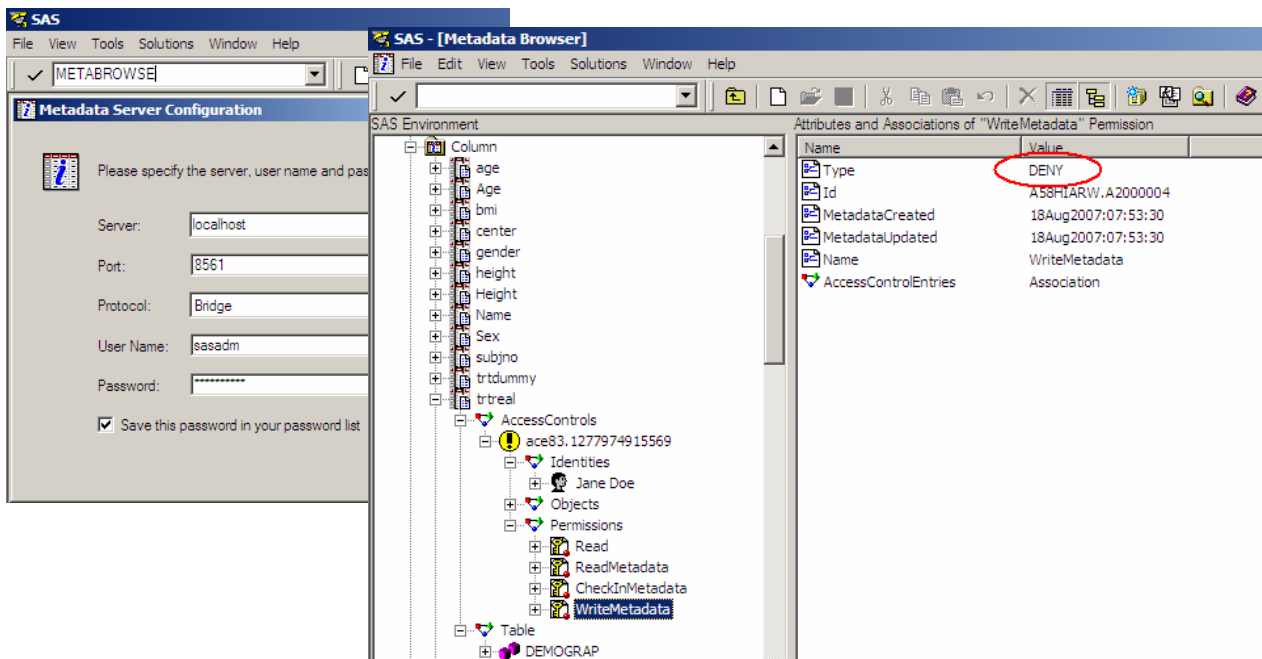
BROWSING METADATA

In the previous example the authorization information stored in the SAS Metadata Server is not accessed or queried, but simply used to make sure Jane Doe cannot read the specific column. In the following examples the PROC METADATA procedure will be used in order to query the metadata. When using the PROC METADATA procedure you will have to know the available metadata types and the hierarchical structure of your metadata.

In order to know what metadata is available for querying and determine the hierarchical structure of your metadata the Metadata Browser can be very helpful.

For Jane Doe we have made specific changes to the Access Controls of the column TRTTREAL in order to make she does not have Read, ReadMetadata, CheckInMetadata and WriteMetadata rights. All these changes are available in the Metadata Browser as is illustrated below. Within interactive SAS, just type 'METABROWSE' in the command bar to invoke the Metadata Browser, and enter the appropriate login information for accessing your metadata server.

PhUSE 2007



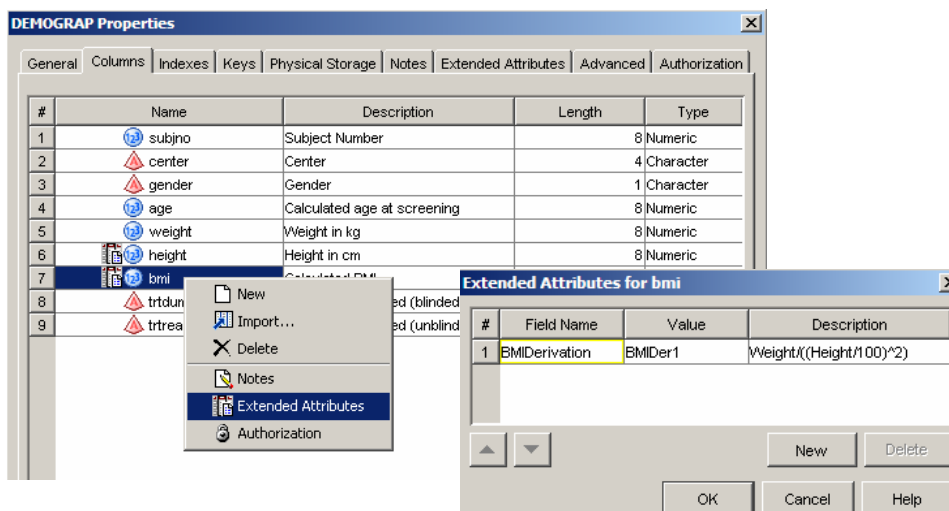
For the column TRTREAL, which is available in the table DEMOGRAP, the access controls are adapted: for identity Jane Doe the permissions Read, ReadMetadata, CheckInMetadata and WriteMetadata are denied.

In the following examples the Metadata Browser can be used to determine the available metadata types and their hierarchical structure. To make sure you see all available metadata types in the Metadata Browser you will have to specify 'Unhide All' in the menu Tools -> Options -> Explorer -> Metadata -> Unhide All.

EXAMPLE 2: QUERYING METADATA AVAILABLE IN EXTENDED ATTRIBUTES

The column BMI in the DEMOGRAP dataset is derived from the columns WEIGHT and HEIGHT. It is derived by dividing the weight (in kilograms) by the square height (in meters). This derivation is straightforward and used globally, but suppose for some strange reason the FDA decides the BMI is to be calculated differently for new studies. Ten years from now you would like to know which studies use the old derivation rule and which studies use the new derivation rule.

Without the use of metadata this would probably be a lot of work, but if the BMI derivation rule was stored in your metadata you could query your metadata and save a lot of time and effort. A logical place to store such a derivation rule would be in the extended attribute of the column in which the outcome of the derivation is stored. In this example we would therefore store the information in the extended attribute of the BMI column.

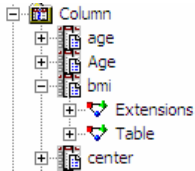


As the BMI derivation is stored in this extended attribute we name this extended attribute 'BMDerivation'. This is the first BMI derivation rule in the company and therefore we specify the value 'BMDer1'. The description of the derivation rule is put in the

PhUSE 2007

description field. After we have entered this information we can extract this information from the SAS Metadata Server by using the PROC METADATA procedure.

Suppose we would like to get an overview of all columns in which the fieldname of the extended attribute contains the text 'Derivation' and we would like to show the attributes of those columns as well as the description of the extended attributes and the table in which the columns are available. With the help of the Metadata Browser we can determine that the extended attributes and tables are associated objects of the Column Type.



Therefore we can create the following PROC METADATA code to query the SAS Metadata server:

```
FILENAME output "C:\phuse\output.xml";
PROC METADATA
  SERVER="localhost"
  PORT=8561
  USERID="sasadm"
  PASSWORD="password"

  IN="<GetMetadataObjects>
    <Reposid>A0000001.A58HIARW</Reposid>
    <Type>Column</Type>
    <Objects/>
    <NS>SAS</NS>
    <Flags>2433</Flags>
    <Options>
      <XMLSelect search="
        Column[Extensions/Extension[@Name CONTAINS 'Derivation']]
      "/>
    </Options>
  </GetMetadataObjects>"
  OUT=output;
RUN;
```

The <GetMetadataObjects> method is used to retrieve metadata objects of a specified metadata type. (This in contrary to the <GetMetadata> method in which a specific object is to be specified and the properties of that specific object are returned.) The <reposid> contains the Metadata ID of our repository (which can be obtained by using the Metadata Browser) and the <type>Column</type>-part specifies all metadata objects of the type Column are to be retrieved. The XMLselect is like a WHERE clause and specifies only columns with an extension that contains in its name the text 'Derivation' are to be kept. The flag '2433' is a so-called IOMI flag which specifies additional requests. The number used here is the sum of 2048, 256, 128 and 1 specifying:

- 1: Gets all of the properties of the requested object and general identifying information about any objects that are associated with the requested object;
- 128: Checks the Options parameter for search criteria that qualifies the objects to return;
- 256: Executes a GetMetadata call for each object that is returned by the GetMetadataObjects method;
- 2048: In GetMetadata, omits all properties that do not contain values or that contain a null value.

The request returns an XML file in which the required information is available:

PhUSE 2007

```

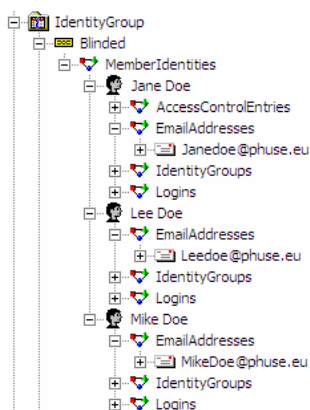
- <GetMetadataObjects>
  <Reposid>A0000001.A58HIARW</Reposid>
  <Type>Column</Type>
  - <Objects>
    - <Column Id="A58HIARW.B70000SQ" Name="height" BeginPosition="0" ColumnLength="8" ColumnName="height" ColumnType="0"
      Desc="Height in cm" EndPosition="0" IsDiscrete="0" IsNullable="1" MetadataCreated="25Aug2007:10:42:55"
      MetadataUpdated="25Aug2007:13:18:34" SASColumnLength="8" SASColumnName="height" SASColumnType="N"
      SASExtendedLength="0" SASPrecision="0" SASScale="0" SummaryRole="STATISTIC">
      - <Extensions>
        <Extension Id="A58HIARW.AB0000RU" Name="HeightDerivation" Desc="1 inch = 2.54 centimeters" />
      </Extensions>
    - <Table>
      <PhysicalTable Id="A58HIARW.B60000RX" Name="DEMOGRAP" />
    </Table>
  </Column>
  - <Column Id="A58HIARW.B70000SR" Name="bmi" BeginPosition="0" ColumnLength="8" ColumnName="bmi" ColumnType="0"
    Desc="Calculated BMI" EndPosition="0" IsDiscrete="0" IsNullable="1" MetadataCreated="25Aug2007:10:42:55"
    MetadataUpdated="25Aug2007:13:18:34" SASColumnLength="8" SASColumnName="bmi" SASColumnType="N"
    SASExtendedLength="0" SASPrecision="0" SASScale="0" SummaryRole="STATISTIC">
    - <Extensions>
      <Extension Id="A58HIARW.AB0000RT" Name="BMIDerivation" Desc="Weight/((Height/100)^2)" />
    </Extensions>
    - <Table>
      <PhysicalTable Id="A58HIARW.B60000RX" Name="DEMOGRAP" />
    </Table>
  </Column>
</Objects>
<NS>SAS</NS>
<Flags>2433</Flags>
- <Options>
  <XMLSelect search="Column[Extensions/Extension[@Name CONTAINS 'Derivation']]" />
</Options>
</GetMetadataObjects>

```

As you can see in the highlighted areas there are two columns available that contain a derivation description in the extended attributes and both columns are available in the DEMOGRAP table. To access this XML file and create for example a SAS dataset with the required information, the SAS XML Mapper or Base SAS can be used.

EXAMPLE 3: EXPLOITING USER AND GROUP INFORMATION

In example 1 we have changed the permissions of the user Jane Doe, but suppose Jane Doe is not the only user that should be blinded. Suppose we have created a group named 'Blinded' and each member of that group should not have permissions to view the column with the unblinded information. We would then simply change the authorization of the group 'blinded' instead of all persons in that group. Suppose we want to send the applicable users an e-mail regarding this change in their authorization. Within the SAS Management Console or the Metadata Browser we can easily determine the users in the 'blinded' group and their e-mail addresses (if they are stored as metadata of course):



However, this information is not in a workable format to automate this process and if there were scores of users in the blinded group it would be labour-intensive to send them all an email. In order to get the information available in a workable format the PROC METADATA procedure can be used:

```

PROC METADATA
  SERVER="localhost"
  PORT=8561
  USERID="sasadm"

```

PhUSE 2007

```

PASSWORD="Password"

IN="<GetMetadataObjects>
  <Reposid>A0000001.A58HIARW</Reposid>
  <Type>Person</Type>
  <Objects/>
  <NS>SAS</NS>
  <Flags>2433</Flags>
  <Options>
    <XMLSelect search=""
      Person[Identitygroups/Identitygroup[@Name = 'Blinded']]
    ""/>
  </Options>
</GetMetadataObjects>"
OUT=output;
RUN;
```

The request returns an XML file in which the required information is available:

```

- <GetMetadataObjects>
  <Reposid>A0000001.A58HIARW</Reposid>
  <Type>Person</Type>
- <Objects>
  - <Person Id="A58HIARW.AP000001" Name="Mike Doe" MetadataCreated="18Aug2007:07:57:52"
    MetadataUpdated="26Aug2007:07:36:37">
    - <EmailAddresses>
      <Email Id="A58HIARW.B9000003" Name="MikeDoe@phuse.eu"/>
    </EmailAddresses>
    - <IdentityGroups>
      <IdentityGroup Id="A58HIARW.A30001JL" Name="Blinded"/>
    </IdentityGroups>
    - <Logins>
      <Login Id="A58HIARW.AQ000001" Name="Login.sasdemo.99"/>
    </Logins>
  </Person>
  - <Person Id="A58HIARW.AP000002" Name="Lee Doe" MetadataCreated="18Aug2007:07:58:36"
    MetadataUpdated="26Aug2007:07:36:23">
    - <EmailAddresses>
      <Email Id="A58HIARW.B9000002" Name="Leedoe@phuse.eu"/>
    </EmailAddresses>
    - <IdentityGroups>
      <IdentityGroup Id="A58HIARW.A30001JL" Name="Blinded"/>
    </IdentityGroups>
    - <Logins>
      <Login Id="A58HIARW.AQ000002" Name="Login.sasadm.8"/>
    </Logins>
  </Person>
  - <Person Id="A58HIARW.AP0000RW" Name="Jane Doe" MetadataCreated="20Aug2007:10:25:26"
    MetadataUpdated="26Aug2007:07:09:20">
    - <AccessControlEntries>
      <AccessControlEntry Id="A58HIARW.A80001JM" Name="ace83.1277974915569"/>
    </AccessControlEntries>
    - <EmailAddresses>
      <Email Id="A58HIARW.B9000001" Name="Janedoe@phuse.eu"/>
    </EmailAddresses>
    - <IdentityGroups>
      <IdentityGroup Id="A58HIARW.A30001JL" Name="Blinded"/>
    </IdentityGroups>
```

The information returned by the SAS Metadata Server can be restricted and expanded by using templates. This will be illustrated in the next example 'Using summary role of columns'.

Of course the data available in this XML file can also be accessed with the SAS XML Mapper or by using Base SAS.

EXAMPLE 4: USING SUMMARY ROLE OF COLUMNS

OCS Consulting has developed a dynamic table creation program which uses the PROC TABULATE procedure in which variables of a certain dataset can be dragged and dropped into a row or a column and calculation variables can be selected:

PhUSE 2007

Portal Page Search

Create a dynamic summary table

Classes

Day

Subject

Dynamic table

Column

Treatment group

Row

Position

Select calculation variables:

☒ Diastolic blood pressure

☒ Heart rate

☒ Systolic blood pressure

Present calculation in:

☒ Column. ☐ Row.

Create Table

In order to prevent users putting a calculation variable (e.g. the systolic blood pressure) in a row or column, or to prevent a variable to be used within this application, different types of variables are defined. This differentiation is not hard coded in the program or in the selection screen, but it is extracted from the metadata.

Within the SAS Management Console for each column Summary Roles are used to specify what kind of variable the column represents. In order to tabulate the DEMOGRAP dataset the following summary roles are set:

DEMOGRAP Properties					
General Columns Indexes Keys Physical Storage Notes Extended Attributes Advanced Authorization					
#	Name	Length	Type	Summary Role	Description
1	subjno	8	Numeric	(None)	Subject Number
2	center	4	Character	CLASS	Center
3	gender	1	Character	CLASS	Gender
4	age	8	Numeric	STATISTIC	Calculated age at screenin
5	weight	8	Numeric	STATISTIC	Weight in kg
6	height	8	Numeric	STATISTIC	Height in cm
7	bmi	8	Numeric	STATISTIC	Calculated BMI
8	trtdummy	20	Character	CLASS	Treatment received (blinde
9	trtreal	20	Character	CLASS	Treatment received (unblin

Within the dynamic table creation program this metadata is extracted via the PROC METADATA procedure to build up the selection screen and assign the variables to their proper function.

In the following example flag 4 is specified indicating a template should be used detailing which items are to be returned:

```
PROC METADATA
  SERVER="localhost"
  PORT=8561
  USERID="sasadm"
  PASSWORD="Password"

  IN='<GetMetadata>
    <Metadata>
      <PhysicalTable Id="A58HIARW.B60000RX" Name="" Desc="" DBMSType=""
        MemberType=""/>
    </Metadata>
    <NS>SAS</NS>
    <Flags>4</Flags>
```


PhUSE 2007

```
<Options>
  <Templates>
    <PhysicalTable Id="" CreatedDT="">
      <Columns/>
    </PhysicalTable>
    <Column Name="" SummaryRole="" />
  </Templates>
</Options>
</GetMetadata>'
OUT=output;
RUN;
```

```
- <GetMetadata>
- <Metadata>
- <PhysicalTable Id="A58HIARW.B60000RX" Name="DEMOGRAP" Desc="" DBMSType="" MemberType="DATA">
  - <Columns>
    <Column Id="A58HIARW.B70000SL" Name="subjno" SummaryRole="" />
    <Column Id="A58HIARW.B70000SM" Name="center" SummaryRole="CLASS" />
    <Column Id="A58HIARW.B70000SN" Name="gender" SummaryRole="CLASS" />
    <Column Id="A58HIARW.B70000SO" Name="age" SummaryRole="STATISTIC" />
    <Column Id="A58HIARW.B70000SP" Name="weight" SummaryRole="STATISTIC" />
    <Column Id="A58HIARW.B70000SQ" Name="height" SummaryRole="STATISTIC" />
    <Column Id="A58HIARW.B70000SR" Name="bmi" SummaryRole="STATISTIC" />
    <Column Id="A58HIARW.B70000SS" Name="trtdummy" SummaryRole="CLASS" />
    <Column Id="A58HIARW.B70000ST" Name="trtreal" SummaryRole="CLASS" />
  </Columns>
</PhysicalTable>
</Metadata>
<NS>SAS</NS>
<Flags>4</Flags>
- <Options>
  - <Templates>
    - <PhysicalTable Id="" CreatedDT="">
      <Columns />
    </PhysicalTable>
    <Column Name="" SummaryRole="" />
  </Templates>
</Options>
</GetMetadata>
- <AccessControlEntries>
  <AccessControlEntry Id="A58HIARW.A80001JM" Name="ace83.1277974915569" />
</AccessControlEntries>
- <EmailAddresses>
  <Email Id="A58HIARW.B9000001" Name="Janedoe@phuse.eu" />
</EmailAddresses>
- <IdentityGroups>
  <IdentityGroup Id="A58HIARW.A30001JL" Name="Blinded" />
</IdentityGroups>
```

CONCLUSION

Metadata can provide meaningful and valuable information. This information can be stored and extracted by using the SAS Open Metadata Architecture. Tools like the Metadata Browser can be very helpful to get an idea of the way your metadata is stored and can therefore help you in creating the appropriate PROC METADATA statements to extract the information needed. Extracted information is available in XML format which can easily be read by SAS and other applications.

REFERENCES

SASOnlineDoc: Getting Started with SAS 9.1.3 Open Metadata Interface

ACKNOWLEDGMENTS

I would like to take the opportunity to thank the following people for contributing to this paper:

Raymond Ebben, OCS Consulting

Yves Poriau, OCS Consulting

RECOMMENDED READING

Examples of how metadata can be of use within the SAS Information Delivery Portal is described in Phuse 2007 paper AD05 "Web-enabling the Pharmaceutical Sector" by Raymond Ebben and Jules van der Zalm, OCS Consulting

SASOnlineDoc: Getting Started with SAS 9.1.3 Open Metadata Interface

PhUSE 2007

CONTACT INFORMATION

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

Bas van Bakel
OCS Consulting
PO BOX 490
5240 AL ROSMALEN
THE NETHERLANDS
Office: +31 (0)73 523 6000
Fax: +31 (0)73 523 6600
sasquestions@ocs-consulting.com
www.ocs-consulting.com

Brand and product names are trademarks of their respective companies.