

PROC INFOMAPS – Let your Design Document build your SAS® V9 Information Maps Chuck Bininger, COMSYS, Kalamazoo, MI

ABSTRACT

Creating information Maps can be a tedious and time consuming process and prone to many errors. Although SAS® Information Map Studio provides a simple GUI interface that allows developers to interactively create Information Maps. It may not be the best tool available.

This paper will introduce the reader to information maps, SAS® Information Map Studio, and PROC INFOMAPS. The paper will also assume that the reader has some knowledge of the SAS® Business Intelligence products. By the end of the paper the reader will be able to understand and use the code provided to dynamically create information maps for use in their own SAS® Business Intelligence environment.

INTRODUCTION

Information maps provide business metadata or descriptors about the data contained within a data warehouse. It is this metadata that provides information to the user about how the data tables are related, how each variable is used in an analysis, and what statistics provide the best mathematical description of the data. When used with SAS Web Report Studio an Information Map provides a transparent view to the data within the data warehouse.

A Data Warehouse requires much attention to detail during the design phase. It is the design that often determines the success or failure of a data warehouse. Design of the Information Maps is equally important. The Information Map acts as the mediator between the Data Warehouse and the Analysts using SAS® Web Report Studio. The Information Map ensures users are querying the data warehouse in a consistent and accurate manner. Information Maps are usually created and managed by information architects and query designers with input from the information consumers. Just as a Data Warehouse changes over time so does an Information Map. It is this change that creates the need for a structured approach of documenting and implementing those changes.

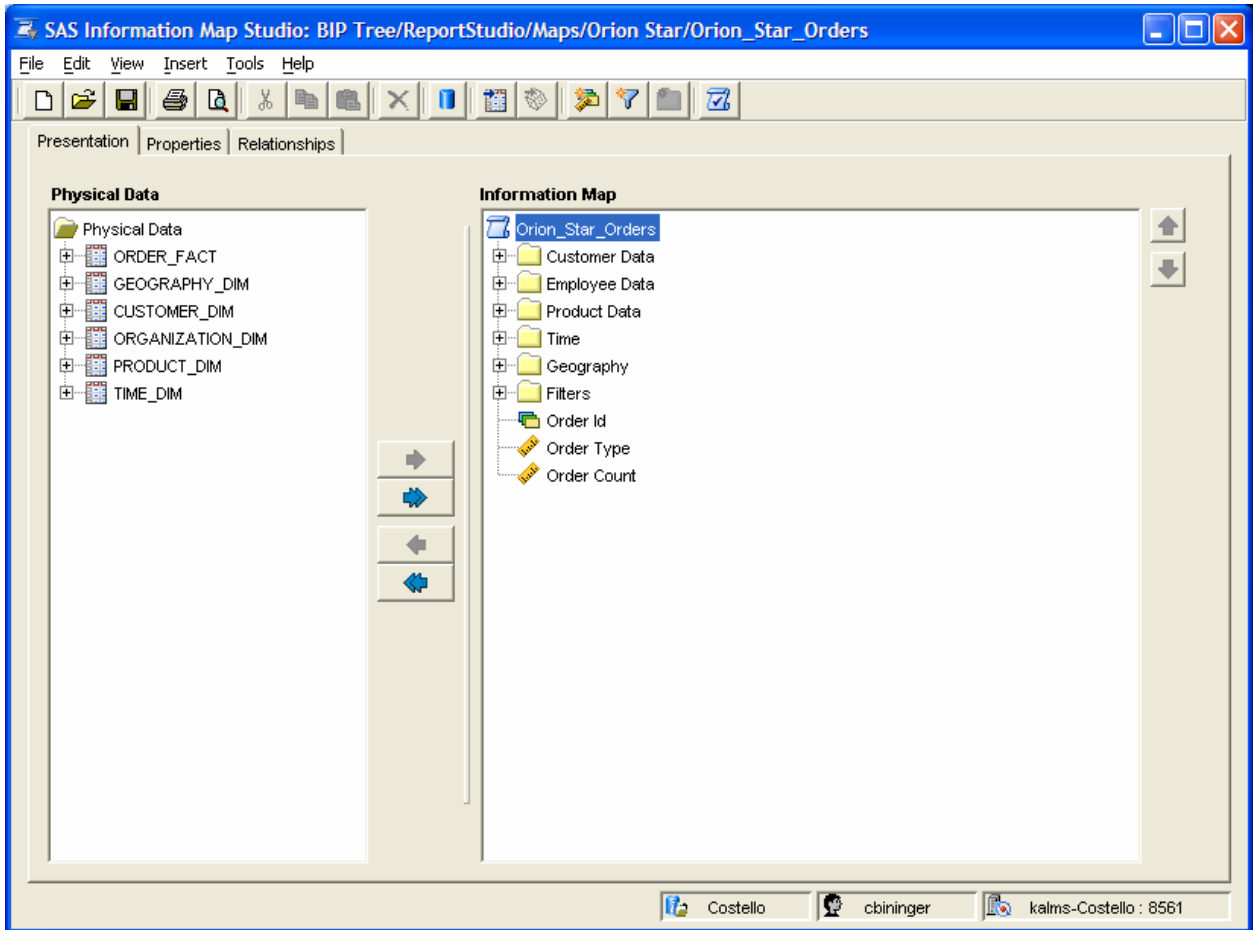
SAS® provides two methods to create information maps.

- SAS® Information Map Studio
- PROC INFOMAPS

SAS® INFORMATION MAP STUDIO

SAS® Information Map Studio provides an easy to use GUI interface that allows Information Architects to produce and manage information maps. SAS® Information Map Studio provides 3 views to an information map;

1. The Presentation tab allows the selection of tables and variables to be included within the information map. This interface also allows the creation of folders for organizing the elements of the information map.



- The Properties tab allows the selection of attributes for each element within the information map. It is here that the user will describe and determine usage of each element.

SAS Information Map Studio: BIP Tree/ReportStudio/Maps/Orion Star/Orion_Star_Orders

File Edit View Insert Tools Help

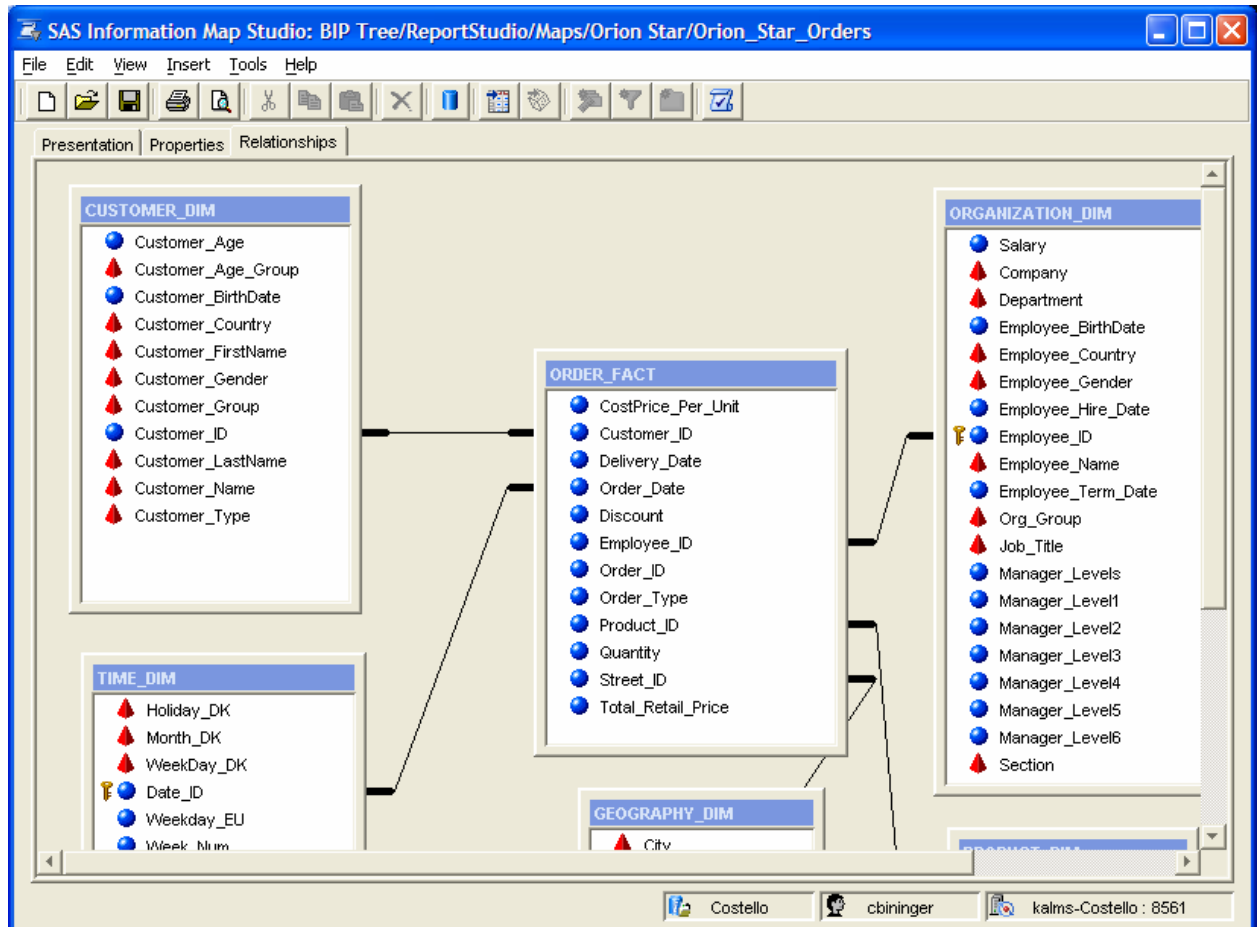
Presentation Properties Relationships

Name	Description
Customer Age	Physical column Customer_Age
Customer Id	Physical column Customer_ID
Customer Age Gr...	Physical column Customer_Age_Group
Customer Birth Date	Physical column Customer_BirthDate
Customer Country	Physical column Customer_Country
Customer First Na...	Physical column Customer_FirstName
Customer Gender	Physical column Customer_Gender
Customer Group ...	Physical column Customer_Group
Customer Last Na...	Physical column Customer_LastName
Customer Name	Physical column Customer_Name
Customer Type N...	Physical column Customer_Type
Annual Salary	Physical column Salary
Employee Id	Physical column Employee_ID
Employee Name	Physical column Employee_Name
Employee Birth Date	Physical column Employee_BirthDate
Employee Country	Physical column Employee_Country
Employee Gender	Physical column Employee_Gender
Employee Hire Date	Physical column Employee_Hire_Date
Employee Termina...	Physical column Employee_Term_Date
Title	Physical column Job_Title
Company	Physical column Company
Department	Physical column Department
Group	Physical column Org_Group
Section	Physical column Section
Product Id	Physical column Product_ID
Product Name	Physical column Product_Name

Property	Value
Data item name	Customer Age
ID	Customer Age
Folder	/Customer Data
Description	Physical column Customer_Age
Expression	...
Data Type	Numeric
Permissions	...
Format	Default format ...
Value generation	Do not provide a list of values (allow user to ...
Classification	Measure ...
Aggregations	SUM(argument) ...

Costello cbininger kalms-Costello : 8561

3. The Relationships tab allows the user to define the join criteria for each table included within the information map.



PROC INFOMAPS

PROC INFOMAPS allows the programmatic creation or modification of information maps.

SYNTAX

```
PROC INFOMAPS
METAPASS="password"
METAPORT=port-number
METAREPOSITORY="repository-name"
METASERVER="address"
METAUSER="user-ID"
<MAPPATH="repository-path">;

DELETE INFOMAP information-map-name
<MAPPATH="repository-path">;

EXPORT <INFOMAP information-map-name>
<FILE=fileref | "physical-path">
<MAPPATH="repository-path">;

IMPORT FILE=fileref | "physical-path";

INSERT DATAITEM
COLUMN=datasource-ID.column-name | EXPRESSION="expression-text"
<AGGREGATION=aggregate-function>
<AGGREGATIONS_DROP_LIST=(aggregate-function-list)>
<CLASSIFICATION=CATEGORY | MEASURE>
<DESCRIPTION="descriptive-text">
<FOLDER=folder-name | "folder-path">
<FORMAT="format-name">
<ID="dataitem-ID">
<NAME="name">
<TYPE=NUMERIC | CHARACTER | DATE | TIME | TIMESTAMP>;

INSERT DATASOURCE
SASSERVER | SERVER=server-name
TABLE=library.table <COLUMNS=(column-1 column-2 ... column-n) | _ALL_ >
<ID=datasource-ID>;

INSERT FILTER filter-name
CONDITION="conditional-expression"
<DESCRIPTION="descriptive-text">
<FOLDER=folder-name| "folder-path">;

INSERT FOLDER folder-name
<PARENT=parent-folder-name>
<DESCRIPTION="descriptive-text">;

INSERT RELATIONSHIP left-table INNER | LEFT | RIGHT | FULL
JOIN right-table ON "conditional-expression";

SAVE <INFOMAP information-map-name>
<MAPPATH="repository-path">;
```

CREATING YOUR DESIGN DOCUMENT

System

Server	Repository	Port	User	Password	Name	MapPath
kalms-Costello	Foundation	8561	cbiningr	Password13	Orion_Star_Orders	/BIP Tree/ReportStudio/Maps/Orion Star

Relationships

Server	SAS Library	Dim Table	Dim ID	Dim Key	Fact Table	Fact ID	Fact Foreign Key
SASMain	Orion Star	Time_Dim	Order_Time_Dim	Date_ID	Order_Fact	Order_Fact	Order_Date
SASMain	Orion Star	Time_Dim	Delivery_Time_Dim	Date_ID	Order_Fact	Order_Fact	Delivery_Date

Columns

Variable	Description	Type	Role	Folder	Default Stat	Table	ID
Customer_ID	Customer ID	Numeric	FK			Order_Fact	Order_Fact
Employee_ID	Employee ID	Numeric	FK			Order_Fact	Order_Fact
Street_ID	Street ID	Numeric	FK			Order_Fact	Order_Fact
Order_Date	Date Order was placed by Customer	Numeric	FK			Order_Fact	Order_Fact
Delivery_Date	Date Order was Delivered	Numeric	FK			Order_Fact	Order_Fact
Order_ID	Order ID	Numeric	Category			Order_Fact	Order_Fact
Order_Type	Order Type	Numeric	Category			Order_Fact	Order_Fact
Product_ID	Product ID	Numeric	FK			Order_Fact	Order_Fact
Quantity	Quantity Ordered	Numeric	Measure	Measures	Sum	Order_Fact	Order_Fact
Total_Retail_Price	Total Retail Price for This Product	Numeric	Measure	Measures	Sum	Order_Fact	Order_Fact
CostPrice_Per_Unit	Cost Price Per Unit	Numeric	Measure	Measures	Sum	Order_Fact	Order_Fact
Discount	Discount in percent of Normal Total Retail Price	Numeric	Measure	Measures	Sum	Order_Fact	Order_Fact
Date_ID	Date of Order	Numeric	Category	Dates		Time_Dim	Order_Time_Dim
Month_Name	Month of Order	Character	Category	Dates		Time_Dim	Order_Time_Dim
Quarter	Quarter of Order	Character	Category	Dates		Time_Dim	Order_Time_Dim
Year_ID	Year of Order	Character	Category	Dates		Time_Dim	Order_Time_Dim
Weekday_Name	Weekday of Order	Character	Category	Dates		Time_Dim	Order_Time_Dim

Filters

Name	Description	Condition	Folder
Weekend Orders	All orders placed on a weekend	<<"Order_time_dim.weekday_of_delivery">> in ('Saturday','Sunday')	Filters

USE THE DESIGN DOCUMENT TO CREATE THE INFORMATION MAP

```
%Macro BuildInfoMap(ExcelFile = ExcelFile);  
  ** Read in Excel file containing definitions **;  
  proc import  
    out = system  
    datafile = "&ExcelFile"  
    dbms = excel2000 replace;  
    range = "System$";  
    getnames = yes;  
  run;  
  
  proc import  
    out = relationships  
    datafile = "&ExcelFile"  
    dbms = excel2000 replace;  
    range = "Relationships$";  
    getnames = yes;  
  run;  
  
  proc import  
    out = columns  
    datafile = "&ExcelFile"  
    dbms = excel2000 replace;  
    range = "Columns$";  
    getnames = yes;  
  run;  
  
  proc import  
    out = filters  
    datafile = "&ExcelFile"  
    dbms = excel2000 replace;  
    range = "Filters$";  
    getnames = yes;
```

```

run;

** Define macro variables based on values from Excel data sets **;
proc sql noprint;
    select server, repository, port, user, password, name, mappath
        into :meta_server, :repository, :port, :user, :password, :name, :mappath
        from system;

    select server, sas_library, fact_table, fact_id
        into :server, :sas_library, :fact_table, :fact_id
        from relationships;

    select dim_table, dim_id, dim_key, fact_foreign_key, count(*)
        into :dim_table1-:dim_table50, :dim_id1-:dim_id50, :dim_key1-:dim_key50,
            :fact_foreign_key1-:fact_foreign_key50, :num_relationships
        from relationships;

    select unique folder, count(unique folder)
        into :folder1-:folder50, :num_folders
        from columns
        where folder ^= '';

    select variable, description, translate(trim(description), '_', ' '),
        format, type, role, folder, default_stat, library, table, ID,
        count(variable)
        into :var1-:var100, :desc1-:desc100, :name1-:name100, :fmt1-:fmt100,
            :type1-:type100, :role1-:role100, :vfolder1-:vfolder100,
            :stat1-:stat100, :lib1-:lib100, :table1-:table100,
            :ID1-:ID100, :num_vfolders
        from columns
        where folder ^= '';

    select server, sas_library, fact_table, fact_id
        into :server, :sas_library, :fact_table, :fact_id

```

```

        from relationships;
quit;

** Build the PROC INFOMAPS code using the macro variables**;
```

```

proc infomaps
    metaserver = "&meta_server"
    metaport = &port
    metarepository = "&repository"
    metauser = "&user"
    metapass = "&password";

    ** Delete any Information Map with the same name **;
    delete infomap "&name" mappath = "&MapPath";

    ** Start building the new Information Map **;
    open infomap "&name" mappath = "&MapPath";

    ** Add the Fact table **;
    insert datasource server="&server"
        table = "&sas_library".&Fact_Table
        ID = "&Fact_ID";

    ** Add the Dimension Tables **;
    %do i = 1 %to &num_relationships;
        insert datasource server="&server"
            table = "&sas_library".&&Dim_table&i
            ID = "&&Dim_ID&i";
    %end;

    ** Define the table relationships **;
    %do i = 1 %to &num_relationships;
        insert relationship
            "&&Dim_ID&i" inner join
            "&Fact_ID" on

```

```

    "(<&&Dim_ID&i...&&Dim_Key&i>>=<&&Fact_ID...&&Fact_Foreign_Key&i>>)" ;
%end;

** Define the Folders **;
%do i = 1 %to &num_folders;
    insert folder "&&folder&i";
%end;

** Add the variables to the folders and define attributes **;
%do i = 1 %to &num_vfolders;
    insert dataitem
        column = "&&ID&i".&&var&i
        id = "&&name&i"
        name = "&&name&i"
        description = "&&desc&i"
        classification = &&role&i
        folder = "&&vfolder&i"
    %if "&&stat&i" ^= "" %then %do;
        aggregation = "&&stat&i"
    %end;
    ;
%end;

** Define the Filters and insert in Filters folder **;
%do i = 1 %to &num_vfolders;
    %if i = 1 %then &do;
        insert folder "Filters";
    %end;
%end;

save; ** Save the Information Map **;
run;
%mend BuildInfoMap;

dm 'cle log'; options mprint maautosource mlogic;

```

<code>%BuildInfoMap(ExcelFile=C:\Sugi\SESUG\Automating Information Maps\Infomap.xls);</code>
--

CONCLUSION

SAS® often provides many tools to accomplish tasks within the SAS Environment. It is the user's responsibility to utilize those tools in a fashion that best fits the needs of the business. This paper has illustrated two ways to create Information Maps. SAS® Information Map Studio provides an easy to use interface to interactively create and test Information Maps. For most development this is the tool of choice!

There are times however when a more formal development approach is preferred. Many times formal design documents are produced and submitted for approval. After approval the design documents are used as instructions to the developers during the implementation phase. It is here that a SAS® program that uses a design document as input and builds Information Map becomes very valuable. As the design changes over time one can update the design document and rerun the program to recreate the Information Map.

Often in SAS® the best solution is often the user's creativity!

REFERENCES

Base SAS® Guide to Information Maps, SAS® Publishing

ACKNOWLEDGMENTS

Special thanks to Chris St. Peter for providing the idea of problematically generating SAS® Information Maps.

RECOMMENDED READING

Base SAS® Guide to Information Maps, SAS® Publishing

SAS® Information Map Studio 3.1: Creating your first Information Map, SAS® Publishing

CONTACT INFORMATION

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

Chuck Bininger
COMSYS
5278 Lovers Lane
Portage, MI 49002
269.344.4100 x549
cbininger@comsys.com
sas.comsys.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.