PharmaSUG 2016 - Paper BB13

# The Power of Data Access Functions: An example with Dataset-XML Creation

Joseph Hinson, inVentiv Health, Princeton, NJ, USA

## ABSTRACT

Data Access Functions are part of the SAS® Component Language or "SCL" (formerly called "Screen Control Language"). Currently, they are not so widely used in clinical programming, except for the OPEN function which is commonly used in SAS® macros when one might need access to data without dealing with a step boundary. But many more functions are available, such as FETCHOBS, VARNUM, and GETVARC. As the name suggests, they provide access to the SAS® data set in a far more flexible way than one can obtain with the SET statement of the Data Step. They are particularly useful in converting tables of dissimilar structures. The SAS® data set is two-dimensional, whereas data structure of the Extensible Markup Language, XML, is hierarchical. Thus converting data sets to XML can be quite challenging. This paper will show that utilizing Data Access Functions can make the conversion of SAS® data sets to XML quite straightforward and easy. The paper uses the creation of the new CDISC Dataset-XML standard as
an example.

## INTRODUCTION

Apparently SAS® programmers have so many tools at their disposal that some tools go unused by and large. A good case in point is the set of data access functions[1], which actually originate from SAS® Component Language, formerly called "Screen Control Language". One exception is the "Open" function (and the corresponding "Close" function) often used by many in SAS® macro programming. A few advanced programmers also use the "ATTRN" function in macro programming to count observations. Compared to the Data Step SET statement, data access functions give much fuller control to the programmer for handling data in a data set. Items in columns and rows are directly accessible as in a 2-dimensional matrix. This allows easier restructuring of data and is particularly useful when converting a two-dimensional data structure ("flat table") like the SAS® data set into a hierarchical form as seen in the extensible markup language, "XML".

The new CDISC document structure for data transport, "dataset-xml" is currently being promoted by the FDA as a potential replacement for the SAS® 5 XPORT (.xpt) format. Dataset-XML[2] is a new standard for data exchange which is a vendor-neutral standard based on CDISC's Operation Data Model (ODM) and therefore compatible with Define-XML while supporting SDTM, ADaM, and SEND CDISC datasets. FDA's recent pilot study[3] has already determined that dataset-XML can transport data and maintain data integrity. Furthermore, the format can handle variable names longer than 8 characters, labels in excess of 40 characters, and text fields greater than 200 characters. The present paper will illustrate how data access functions can be used to easily compose dataset-XML from SAS® data sets.

## DATA ACCESS FUNCTIONS

Data Access Functions originate from a SAS® scripting language called SAS® Component Language (SCL). SCL was created for interactive SAS® applications and is the scripting language underlying SAS®/AF, SAS®/FSP, and SAS®/EIS applications. About a dozen SCL functions are available for use in regular BASE SAS® programs:

| SCL Function | Utility in Regular BASE SAS® |
|---|---|
| OPEN | Opens a SAS® data set and assigns an ID number (dsid) which is to refer to that particular data set and is used by the functions below. |
| FETCH | Retrieves the next data set observation and inserts it into the Dataset Data Vector, DDV, which is analogous to the PDV. |
| FETCHOBS | Retrieves the specified data set observation and insert it into the DDV |
| GETVARN | Retrieves the numeric value of the specified variable from the DDV and assign it to a program variable |
| GETVARC | Retrieves the character value of the specified variable from the DDV and assign it to a program variable |
| VARNUM | Gets the position of a variable in a SAS® data set given a variable name. |
| VARTYPE | Returns the corresponding variable type given the variable's position in a data set. |
| VARFMT | Returns the corresponding variable format given the variable's position in a data set. |
| VARINFMT | Returns the corresponding variable informat given the variable's position in a data set. |
| VARLABEL | Returns the corresponding variable label given the variable's position in a data set. |
| VARLEN | Returns the corresponding variable length given the variable's position in a data set. |
| ATTRN or ATTRC | Retrieves the value from the data set descriptor info for the specified attribute. Eg. ATTRN (dsid, "NOBS") for number of observations, ATTRN (dsid, "NVARS") for number of columns. |
| CLOSE | Closes that particular data set opened with OPEN. Eg "CLOSE (dsid)" |

**Table 1. Data Access Functions Usable in Base SAS®**

## APPLICATION TO CDISC DATASET-XML CREATION

According to CDISC, the purpose of Dataset-XML is to support the interchange of tabular clinical research data using Operational Data Model (ODM) XML technologies. The ODM model is vendor-neutral and platform independent and includes the clinical data along with its associated metadata, administrative data, reference data and audit information.

The data exchanged or transmitted using the Dataset-XML format is expected to match the metadata definitions provided in another ODM-based document, the *define.xml,* which accompanies the dataset-XML file. Thus, a set of Dataset-XML files would be expected to represent the data for one clinical study and the metadata for those Dataset-XML files would be provided by the accompanying Define-XML file.

In addition to supporting the transport of datasets for submission to the FDA, the Dataset-XML format may also be used to facilitate other data interchange use, such as the transmission of SDTM, ADaM, or SEND CDISC datasets to an organization.

As previously mentioned, the CDISC dataset-xml document structure is based on the Operational Data Model (ODM) and therefore has the following key components:

- the XML header element (indicates the beginning of an XML file)
  ```
  <?xml version="1.0" encoding="UTF-8" ?>'
  ```
- the root ODM element (including Study information)

```
-<ODM CreationDateTime="2012-12-08T23:06:03.046+01:00" FileOID="SDTM-XML"
FileType="Snapshot" ODMVersion="1.3.2" data:DatasetXMLVersion="1.0.0"
xmlns="http://www.cdisc.org/ns/odm/v1.3"
xmlns:data="http://www.cdisc.org/ns/Dataset-XML/v1.0">
```

- the ClinicalData or ReferenceData element depending on dataset contents

```
-<ClinicalData MetaDataVersionOID="CDISC.SDTM.3.1.0" StudyOID="LZZT">
```

- ItemGroupData element for each observation, with a unique data:ItemGroupDataSeq attribute representing the data set record number (Naming convention for ItemGroupOID: IG)

```
<ItemGroupData ItemGroupOID="AE" data:ItemGroupDataSeq="1">
```

- ItemData element for each non-missing data value within a record. (Naming convention for ItemOID: IT)

```
<ItemData Value="CDISCPILOT01" ItemOID="AE.STUDYID"/>
```

## PROGRAMMING STRATEGY

The conversion of an SDTM SAS® data set to dataset-XML is carried out as follows:

1. Create an SDTM data set from raw data.
2. Begin generating dataset-XML file by putting out header elements.
3. Open SDTM data set and obtain a data set ID number with the OPEN function.
4. Obtain the number of rows and number of columns with the ATTRN function.
5. Use the number of rows and columns as indices in a DO loop.
6. Using the FETCHOBS function, read each observation and loop through the columns.
7. While looping through the rows and columns, use the VARNAME function to pick up variable names. Also use the VARTPYE function to determine variable type.
8. Based on variable type, use GETVARN or GETVARC functions to pick up variable values.
9. Use the information obtained with VARNAME, GETVARN, and GETVARC to compose XML elements for output.

Use of Data Access Functions in the Code:

OPEN DATA SET:

```
dsid=open("name of dataset");
```

GET NUMBERS OF OBSERVATIONS AND COLUMNS:

```
nrows=attrn(dsid,'NOBS');
ncols=attrn(dsid,'NVARS');
```

USE ATTRN (NOBS) VALUES TO LOOP THROUGH ROWS:

```
do rx=1 to nrows;
        -----
        -----
```

GET ONE OBSERVATION OF DATA:

```
        rf=fetchobs(dsid,rx);
```

USE ATTRN (NVARS) VALUES TO LOOP THROUGH COLUMNS:

```
        do cx=1 to ncols;
```

GET NAME OF VARIABLE:

```
                xitem=varname(dsid,cx);
```

GET VARIABLE TYPE (to use for choosing GETVARN or GETVARC functions):

```
                xtype=vartype(dsid,cx);
```

GET VARIABLE VALUE BASED ON TYPE:

```
                    if xtype eq 'N' then xvalue=put(getvarn(dsid, cx), 8.);
                    else if xtype eq 'C' then xvalue=getvarc(dsid, cx);
```

COMPOSE AN XML ELEMENT USING DATA OBTAINED WITH VARNAME(), GETVARN(), OR GETVARC():

```
xstring='<ItemData ItemOID="IT.'||strip(xitem)||'" Value="'||strip(xvalue)||'" />';
put xstring;
```

This is the paper body. This is another sample paragraph. This is another sample paragraph. This is another sample paragraph. This is another sample paragraph. This is another sample paragraph. This is another sample paragraph. This is another sample paragraph. This is another sample paragraph. This is another sample paragraph.

## THE COMPLETE SAS® CODE:

This is subtopic for the above. This is the paper body. This is the paper body. This is the paper body. This is the paper body. This is the paper body. This is the paper body. This is the paper body.

Output 1 shows an example of how to present output.

(a) Data Set:

Create AE data set from raw data:

```
data aetable;
        infile datalines dsd dlm=',' n=500 missover;
        input STUDYID $ DOMAIN $ USUBJID :$30. AESEQ AESPID $ AETERM :$30. AEMODIFY :$30. AEDECOD
:$30.
        AEBODSYS :$50. AESEV :$10. AESER $ AEACN :$20. AEREL :$20. AESTDTC :$20. AEENDTC :$20.
AESTDY AEENDY AEENRF $ ;
datalines;
CDISC01, AE, CDISC01.100008, 1, 1, AGITATED, AGITATION, Agitation, Psychiatric disorders, MILD,
N, DOSE NOT CHANGED, POSSIBLY RELATED, 2003-05,, 3,, AFTER
CDISC01, AE, CDISC01.100008, 2, 2, ANXIETY,, Anxiety, Psychiatric disorders, MODERATE, N,  DOSE
NOT CHANGED, POSSIBLY RELATED, 2003-05-13,, 15,, AFTER
CDISC01, AE, CDISC01.100008, 3, 3, DECREASED APPETITE,, Decreased appetite, Metabolism and
nutrition disorders, MILD, N, DOSE NOT CHANGED, POSSIBLY RELATED, 2003-08-19, 2003-09-15, 113,
140,
CDISC01, AE, CDISC01.100014, 1, 1, DIARRHEA,, Diarrhoea, Gastrointestinal disorders, MILD, N,
DOSE NOT CHANGED, NOT RELATED, 2004-01-06,, 84,, AFTER
CDISC01, AE, CDISC01.100014, 2, 2, HEMORRHOIDS,, Haemorrhoids, Gastrointestinal disorders,
MODERATE, N, DOSE NOT CHANGED, NOT RELATED, 2004-01-06,, 84,, AFTER
CDISC01, AE, CDISC01.100014, 3, 3, HEADACHE,, Headache, Nervous system disorders, MILD, N, DOSE
NOT CHANGED, NOT RELATED, 2004-01-27,, 105,,AFTER
CDISC01, AE, CDISC01.100014, 4, 4, VOMIT, VOMITING, Vomiting, Gastrointestinal disorders,
MODERATE, N, DRUG INTERRUPTED, POSSIBLY RELATED, 2004-02-03, 2004-02-03, 112,112,
CDISC01, AE, CDISC01.100014, 5, 5, VOMIT, VOMITING, Vomiting, Gastrointestinal disorders, SEVERE,
Y, DRUG INTERRUPTED, POSSIBLY RELATED, 2004-02-04, 2004-02-09, 113,118,
CDISC01, AE, CDISC01.200001, 1, 1, ANXIETY,, Anxiety, Psychiatric disorders, SEVERE, N, DOSE NOT
CHANGED, POSSIBLY RELATED, 2003-10-16, 2003-10-20, 17, 21,
CDISC01, AE, CDISC01.200001, 2, 5, LEFT KNEE PAIN WORSENING,, Arthralgia, Musculoskeletal and
connective tissue disorders,  SEVERE, N, DRUG WITHDRAWN,  NOT RELATED, 2004-02-02,, 126,, AFTER
CDISC01, AE, CDISC01.200001, 3, 3, CONSTIPATION,, Constipation, Gastrointestinal disorders,
MODERATE, N, DOSE NOT CHANGED,  NOT RELATED,  2003-12-25,, 87,,AFTER
CDISC01, AE, CDISC01.200001, 4, 4, TIREDNESS,, Fatigue, General disorders and administration site
conditions, SEVERE, N, DOSE NOT CHANGED, POSSIBLY RELATED, 2003-12-25,,  87,, AFTER
CDISC01, AE, CDISC01.200001, 5, 2, NAUSEA INTERMITTENT,, Nausea, Gastrointestinal disorders,
SEVERE, N, DOSE NOT CHANGED, POSSIBLY RELATED, 2003-10-16, 2003-10-20, 17,21,
CDISC01, AE, CDISC01.200002, 1, 3, LIGHTHEADEDNESS,,Dizziness, Nervous system disorders, MILD, N,
DOSE NOT CHANGED,  NOT RELATED,2004-02-26, 2004-02-26, 140, 140,
CDISC01, AE, CDISC01.200002, 2, 1, MUSCLE SPASMS,, Muscle spasms, Musculoskeletal and connective
tissue disorders,  MILD, N, DOSE NOT CHANGED, NOT RELATED, 2004-01-05,, 88,, AFTER
CDISC01, AE, CDISC01.200002, 3, 2,PALPITATIONS INTERMITTENT,, Palpitations, Cardiac disorders,
MILD, N, DOSE NOT CHANGED,  NOT RELATED, 2004-01-05,, 88,, AFTER
;
run;
```

(b) Set Up XML Output:

```
%let ds=aetable;
%let xmlpath=%str(C:\Users\admin\Desktop\SASoutputs\&ds..xml);

filename xmlout "&xmlpath";
```

(c) Compute Current Date/Time Macro Variable for Insertion into XML Header:

```
data _null_;
        length isodtmx $19;
        isodtm=datetime();
        isodtmx=put(isodtm,e8601dt19.);
        call symputx('isodtmc',isodtmx);
run;
```

(d) Use Data Access Functions to Read Data and Compose Dataset-XML elements:

```
options lrecl=10000;

data _null_;
  length xstring tstring $250 xitem rxc $15 xvalue $30;
  file xmlout;
  dsid=open("&ds");
  nrows=attrn(dsid,'NOBS');
  ncols=attrn(dsid,'NVARS');
  %let isodatetime=&isodtmc;
  tstring="CreationDateTime="||byte(34)||"&isodatetime."||byte(34)||byte(62);
  xstring='<?xml version="1.0" encoding="UTF-8" ?>';
  put xstring;
  put '<ODM xmlns="http://www.cdisc.org/ns/odm/v1.3" xmlns:xlink="http://www.w3.org/1999/xlink"';
  put  'xmlns:data="http://www.cdisc.org/ns/Dataset-XML/v1.0" FileType="Snapshot"
ODMVersion="1.3.2" ';
  put 'data:DatasetXMLVersion="1.0.0" FileOID="www.cdisc.org.Studycdisc01-Define-
XML_2.0.0(IG.AE)"';
  xstring='PriorFileOID="www.cdisc.org.Studycdisc01-Define-XML_2.0.0" Originator="SAS Programmer
Joe Hinson" '||tstring;
  put xstring;

  put  "<ClinicalData>";
  do rx=1 to nrows;
    rxc=strip(put(rx,8.));
    rf=fetchobs(dsid,rx);
    xstring= '<ItemGroupData ItemGroupOID="IG.AE" data:ItemGroupDataSeq="'||rxc||'" >';
    put xstring;

    do cx=1 to ncols;
      xitem=varname(dsid,cx);
      xtype=vartype(dsid,cx);
      if xtype eq 'N' then xvalue=put(getvarn(dsid, cx), 8.);
      else if xtype eq 'C' then xvalue=getvarc(dsid, cx);
      xstring='<ItemData ItemOID="IT.'||strip(xitem)||'" Value="'||strip(xvalue)||'" />';
      put xstring;
    end;
    put   "</ItemGroupData>";
  end;
  put  "</ClinicalData>";
  put "</ODM>";

  stop;
run;
```

**Output 1. Output from a CREATE TABLE Statement**

```
<?xml version="1.0" encoding="UTF-8"?>
- <ODM CreationDateTime="2016-01-02T18:26:14" Originator="SAS Programmer Joe Hinson" PriorFileOID="www.cdisc.org.Studycdisc01-Define-XML_2.0.0"
  FileOID="www.cdisc.org.Studycdisc01-Define-XML_2.0.0(IG.AE)" data:DatasetXMLVersion="1.0.0" ODMVersion="1.3.2" FileType="Snapshot"
  xmlns:data="http://www.cdisc.org/ns/Dataset-XML/v1.0" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.cdisc.org/ns/odm/v1.3">
   - <ClinicalData>
      - <ItemGroupData data:ItemGroupDataSeq="1 " ItemGroupOID="IG.AE">
            <ItemData Value="CDISC01" ItemOID="IT.STUDYID"/>
            <ItemData Value="AE" ItemOID="IT.DOMAIN"/>
            <ItemData Value="CDISC01.100008" ItemOID="IT.USUBJID"/>
            <ItemData Value="1" ItemOID="IT.AESEQ"/>
            <ItemData Value="1" ItemOID="IT.AESPID"/>
            <ItemData Value="AGITATED" ItemOID="IT.AETERM"/>
            <ItemData Value="AGITATION" ItemOID="IT.AEMODIFY"/>
            <ItemData Value="Agitation" ItemOID="IT.AEDECOD"/>
            <ItemData Value="Psychiatric disorders" ItemOID="IT.AEBODSYS"/>
            <ItemData Value="MILD" ItemOID="IT.AESEV"/>
            <ItemData Value="N" ItemOID="IT.AESER"/>
            <ItemData Value="DOSE NOT CHANGED" ItemOID="IT.AEACN"/>
            <ItemData Value="POSSIBLY RELATED" ItemOID="IT.AEREL"/>
            <ItemData Value="2003-05" ItemOID="IT.AESTDTC"/>
            <ItemData Value="" ItemOID="IT.AEENDTC"/>
            <ItemData Value="3" ItemOID="IT.AESTDY"/>
            <ItemData Value="." ItemOID="IT.AEENDY"/>
            <ItemData Value="AFTER" ItemOID="IT.AEENRF"/>
        </ItemGroupData>
      - <ItemGroupData data:ItemGroupDataSeq="2 " ItemGroupOID="IG.AE">
            <ItemData Value="CDISC01" ItemOID="IT.STUDYID"/>
            <ItemData Value="AE" ItemOID="IT.DOMAIN"/>
            <ItemData Value="CDISC01.100008" ItemOID="IT.USUBJID"/>
            <ItemData Value="2" ItemOID="IT.AESEQ"/>
            <ItemData Value="2" ItemOID="IT.AESPID"/>
            <ItemData Value="ANXIETY" ItemOID="IT.AETERM"/>
            <ItemData Value="" ItemOID="IT.AEMODIFY"/>
            <ItemData Value="Anxiety" ItemOID="IT.AEDECOD"/>
            <ItemData Value="Psychiatric disorders" ItemOID="IT.AEBODSYS"/>
            <ItemData Value="MODERATE" ItemOID="IT.AESEV"/>
            <ItemData Value="N" ItemOID="IT.AESER"/>
            <ItemData Value="DOSE NOT CHANGED" ItemOID="IT.AEACN"/>
            <ItemData Value="POSSIBLY RELATED" ItemOID="IT.AEREL"/>
            <ItemData Value="2003-05-13" ItemOID="IT.AESTDTC"/>
            <ItemData Value="" ItemOID="IT.AEENDTC"/>
            <ItemData Value="15" ItemOID="IT.AESTDY"/>
            <ItemData Value="." ItemOID="IT.AEENDY"/>
            <ItemData Value="AFTER" ItemOID="IT.AEENRF"/>
        </ItemGroupData>
      - <ItemGroupData data:ItemGroupDataSeq="3 " ItemGroupOID="IG.AE">
            <ItemData Value="CDISC01" ItemOID="IT.STUDYID"/>
            <ItemData Value="AE" ItemOID="IT.DOMAIN"/>
            <ItemData Value="CDISC01.100008" ItemOID="IT.USUBJID"/>
```

**Output 2. The Top Part of the Dataset-XML for AE**

```
            <ItemData Value="2004-02-26" ItemOID="IT.AESTDTC"/>
            <ItemData Value="2004-02-26" ItemOID="IT.AEENDTC"/>
            <ItemData Value="140" ItemOID="IT.AESTDY"/>
            <ItemData Value="140" ItemOID="IT.AEENDY"/>
            <ItemData Value="" ItemOID="IT.AEENRF"/>
        </ItemGroupData>
      - <ItemGroupData data:ItemGroupDataSeq="15 " ItemGroupOID="IG.AE">
            <ItemData Value="CDISC01" ItemOID="IT.STUDYID"/>
            <ItemData Value="AE" ItemOID="IT.DOMAIN"/>
            <ItemData Value="CDISC01.200002" ItemOID="IT.USUBJID"/>
            <ItemData Value="2" ItemOID="IT.AESEQ"/>
            <ItemData Value="1" ItemOID="IT.AESPID"/>
            <ItemData Value="MUSCLE SPASMS" ItemOID="IT.AETERM"/>
            <ItemData Value="" ItemOID="IT.AEMODIFY"/>
            <ItemData Value="Muscle spasms" ItemOID="IT.AEDECOD"/>
            <ItemData Value="Musculoskeletal and connective" ItemOID="IT.AEBODSYS"/>
            <ItemData Value="MILD" ItemOID="IT.AESEV"/>
            <ItemData Value="N" ItemOID="IT.AESER"/>
            <ItemData Value="DOSE NOT CHANGED" ItemOID="IT.AEACN"/>
            <ItemData Value="NOT RELATED" ItemOID="IT.AEREL"/>
            <ItemData Value="2004-01-05" ItemOID="IT.AESTDTC"/>
            <ItemData Value="" ItemOID="IT.AEENDTC"/>
            <ItemData Value="88" ItemOID="IT.AESTDY"/>
            <ItemData Value="." ItemOID="IT.AEENDY"/>
            <ItemData Value="AFTER" ItemOID="IT.AEENRF"/>
        </ItemGroupData>
      - <ItemGroupData data:ItemGroupDataSeq="16 " ItemGroupOID="IG.AE">
            <ItemData Value="CDISC01" ItemOID="IT.STUDYID"/>
            <ItemData Value="AE" ItemOID="IT.DOMAIN"/>
            <ItemData Value="CDISC01.200002" ItemOID="IT.USUBJID"/>
            <ItemData Value="3" ItemOID="IT.AESEQ"/>
            <ItemData Value="2" ItemOID="IT.AESPID"/>
            <ItemData Value="PALPITATIONS INTERMITTENT" ItemOID="IT.AETERM"/>
            <ItemData Value="" ItemOID="IT.AEMODIFY"/>
            <ItemData Value="Palpitations" ItemOID="IT.AEDECOD"/>
            <ItemData Value="Cardiac disorders" ItemOID="IT.AEBODSYS"/>
            <ItemData Value="MILD" ItemOID="IT.AESEV"/>
            <ItemData Value="N" ItemOID="IT.AESER"/>
            <ItemData Value="DOSE NOT CHANGED" ItemOID="IT.AEACN"/>
            <ItemData Value="NOT RELATED" ItemOID="IT.AEREL"/>
            <ItemData Value="2004-01-05" ItemOID="IT.AESTDTC"/>
            <ItemData Value="" ItemOID="IT.AEENDTC"/>
            <ItemData Value="88" ItemOID="IT.AESTDY"/>
            <ItemData Value="." ItemOID="IT.AEENDY"/>
            <ItemData Value="AFTER" ItemOID="IT.AEENRF"/>
        </ItemGroupData>
    </ClinicalData>
</ODM>
```

**Output 2. The Bottom Part of the Dataset-XML for AE**

## CONCLUSION

The paper has demonstrated the use of Data Access Functions for composing Dataset-XML elements from an SDTM data set. The functions are capable of many other SAS® manipulations such as table transpositions[4] and the creation of Patient Profiles (not shown). Potentially, any SAS® data processing involving complex table manipulations can benefit from such functions.

## REFERENCES

1. Galbis-Reig, Felix. “Data Without (Step) Boundaries: Using Data Access Functions”. Proceedings of The NorthEast SAS® Users Group, 2007 Baltimore, MD, USA

http://www.lexjansen.com/nesug/nesug07/cc/cc15.pdf

2. CDISC, “Dataset-XML”

http://www.cdisc.org/dataset-xml

3. The US Food and Drug Administration,”Test Report for DS-XML Pilot”, April 8, 2015.

http://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM443327.pdf

4. Hinson, Joseph and Shi, Changhong. “Transposing Tables from Long to Wide: A Novel Approach Using Hash Objects”. PharmaSUG 2012, Paper PO14.

http://www.pharmasug.org/proceedings/2012/PO/PharmaSUG-2012-PO14.pdf

## ACKNOWLEDGMENTS

This author is very grateful to his manager, mentor, and teacher Brian Shilling, for his constant guidance and ready assistance.

## CONTACT INFORMATION

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

Joseph W. Hinson, PhD
inVentiv Health
202 Carnegie Center, Suite 200
Princeton, NJ, 08540
1-609-282-1615
joehinson@outlook.com


# Page of Formatted Samples

This page has samples that you can copy into the body of your paper and adapt as necessary for your content.

Note: Delete this page before submitting your paper.

**Source Code Sample**

```
data one;
set two;
if mix(var1, var2) > 0 then do;
```

**List: Numbered or Ordered**

1. numbered list item
2. numbered list item
3. numbered list item

**List: Bulleted or Unordered**

- This is a sample bulleted list item.
- This is a sample bulleted list item.

**Output Sample**

```
CREATE TABLE ALLACCTX(SourceSystem varchar(4),
cctnum numeric(18,5) CONSTRAINT "ALLACCT_PK" PRIMARY KEY,
ccttype numeric(18,5),balance numeric(18,5),clientid numeric(18,5),
losedate date,opendate date,primary_cd numeric(18,5),status varchar(1))
```

**Output 3. Output from a CREATE TABLE Statement**

**Table Sample**

| Heading for Column 1 | Heading for Column 2 | Heading for Column 3 | Heading for Column 4 |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

**Table 1. Sample Table**

# Basic Instructions to Insert Captions, Cross-References, and Graphics

These instructions are written for MS Word 2007 and 2010. The steps are similar for MS Word 2003.

To insert a caption:

1. Click References on the main Word menu.
2. Click Insert Caption.
3. Select the Label type you want.
4. Click OK.

To insert a cross-reference:

1. Click References on the main Word menu.
2. Click Cross-reference.
3. In the Reference type list box, select Figure, Table, Display, or Output.

4. In the For which caption list, select the caption you want.
5. From the Insert reference to list, select Only label and number.

To insert a graphic from a file:

1. Click Insert on the main Word menu.
2. Click Picture.
3. In the Insert Picture dialog box, navigate to the file you want to insert.
4. When the name of the file you want to insert is displayed in the File name box, click Insert.