

Using XML Mapper and Enterprise Guide to Read Data and Metadata from an XML File

Larry Hoyle, Inst. For Policy & Social Research

University of Kansas

Kansas City SAS User Group, March 2011

General Issue

- Read an XML file containing both
 - Data
 - Metadata
- Convert those data into something usable in SAS

Why Worry About Metadata?

**Here are
your data.**

**What's
the
problem?**

Employment Manager	11	17	1	
Assistant Prog Manager	2	2	3	
Director of Laboratories	2	2	4	
Project Leader	15	21	2	
Information Technology Anaylst		2	2	2
Computer Operations Manager		1	1	2

```
H000000114200010080999900073002201000002006122
P00000010000013220025000000010400000000501020
H000000214200010080999900038002802000002006222
P00000020000016000029000000010100000000501020
P000000201100156000320000000231040000000501048
H000000314200010080999900020002501000002005322
P00000030000013020029000000010400000000404020
```

So What Comprises Metadata?

- Resource Discovery –
 - Author, Location, Version, Date, Description, Rights, Citations/Uses ...
- “Codebook” –
 - Variables, Code Schemes, Physical Layout
- Concepts, Categories
- Context –
 - Universe, Geography, Time
- Collection Methods
- Processing Methods
- Related Data

• Find the data

• Use the data

• Interpret the data

Specific Goal for This Example: Stata “dta” XML file To SAS Dataset

- Data
- SAS compatible metadata (Labels, formats)
- Other metadata

Our XML Example

```
1 <?xml version="1.0" encoding="US-ASCII" standalone="yes"?>
2 <!DOCTYPE dta [
3 <!ELEMENT dta (header, descriptors,
4 variable_labels, expansion,
5 data, value_labels)>
6 <!ELEMENT header (ds_format, byteorder,
7 filetype, nvar, nobs, data_label,
8 time_stamp)>
9 <!ELEMENT ds_format (#PCDATA)>
10 <!ELEMENT byteorder (#PCDATA)>
11 <!ELEMENT filetype (#PCDATA)>
12 <!ELEMENT nvar (#PCDATA)>
13 <!ELEMENT nobs (#PCDATA)>
14 <!ELEMENT data_label (#PCDATA)>
15 <!ELEMENT time_stamp (#PCDATA)>
16 <!ELEMENT descriptors (typelist, varlist,
17 srtlist, fmtlist, lblist)>
18 <!ELEMENT typelist (type)+>
19 <!ELEMENT type (#PCDATA)>
20 <!ATTLIST type varname IDREF #REQUIRED>
21 <!ELEMENT varlist (variable)+>
22 <!ELEMENT variable EMPTY>
23 <!ATTLIST variable varname ID #REQUIRED>
24 <!ELEMENT srtlist (sort)*>
25 <!ELEMENT sort EMPTY>
26 <!ATTLIST sort varname IDREF #REQUIRED>
27 <!ELEMENT fmtlist (fmt)+>
28 <!ELEMENT fmt (#PCDATA)>
```

Begins with a Document Type Definition (DTD)

Defines root element as <dta>

Defines the rest of the elements and how they nest

Our XML Example - 2

```
1 <?xml version="1.0" encoding="US-ASCII" standalone="yes"?>
2 <!DOCTYPE dta [
3 <!ELEMENT dta (header, descriptors,
4 variable_labels, expansion,
5 data, value_labels)>
6 <!ELEMENT header (ds_format, byteorder,
```

Begins with a Document Type Definition (DTD)

Root element is <dta>

```
50 ]>
51 <dta>
52 <header>
53 <ds_format>113</ds_format>
54 <byteorder>LOHI</byteorder>
55 <filetype>1</filetype>
56 <nvar>11</nvar>
57 <nobs>34</nobs>
58 <data_label>Data on SAS International Conference</data_label>
59 <time_stamp>28 Dec 2009 10:33</time_stamp>
60 </header>
61 <descriptors>
62 <typelist>
63 <type varname='name'>str21</type>
64 <type varname='confurl'>str60</type>
65 <type varname='location'>str45</type>
66 <type varname='city'>str16</type>
67 <type varname='state'>str12</type>
```

Data types

Our XML Example - 3

```
88 <srtlist>
89 <sort varname='' />
90 <sort varname='' />
91 <sort varname='' />
92 </srtlist>
93 <fmtlist>
94 <fmt varname='name'>%21s</fmt>
95 <fmt varname='confurl'>%60s</fmt>
96 <fmt varname='location'>%45s</fmt>
97 <fmt varname='city'>%16s</fmt>
98 <fmt varname='state'>%12s</fmt>
99 <fmt varname='chair'>%32s</fmt>
100 <fmt varname='attendees'>%8.0g</fmt>
101 <fmt varname='proceedingsurl'>%62s</fmt>
102 <fmt varname='startdate'>%td</fmt>
103 <fmt varname='enddate'>%td</fmt>
104 <fmt varname='OnLexJansen'>%8.0g</fmt>
105 </fmtlist>
106 <lbllist>
107 <lblname varname='name'></lblname>
108 <lblname varname='confurl'></lblname>
109 <lblname varname='location'></lblname>
110 <lblname varname='city'></lblname>
111 <lblname varname='state'></lblname>
112 <lblname varname='chair'></lblname>
113 <lblname varname='attendees'></lblname>
114 <lblname varname='proceedingsurl'></lblname>
115 <lblname varname='startdate'></lblname>
116 <lblname varname='enddate'></lblname>
117 <lblname varname='OnLexJansen'>OnJansen</lblname>
118 </lbllist>
```

Sort order

Display Formats

Value label assignments

Our XML Example - 4

```
119 </descriptors>
120 <variable_labels>
121 <vlabel varname='name'>Conference Name</vlabel>
122 <vlabel varname='confurl'>URL of Conference Web Site</vlabel>
123 <vlabel varname='location'>Conference Location</vlabel>
124 <vlabel varname='city'>Conference City</vlabel>
125 <vlabel varname='state'>Conference State</vlabel>
126 <vlabel varname='chair'>Conference Chair(s)</vlabel>
127 <vlabel varname='attendees'>Number of Attendees</vlabel>
128 <vlabel varname='proceedingsurl'>URL of Proceedings</vlabel>
129 <vlabel varname='startdate'>First Day of Conference</vlabel>
130 <vlabel varname='enddate'>Last Day of Conference</vlabel>
131 <vlabel varname='OnLexJansen'>Papers Available on lexjansen.c
132 </variable_labels>
133 <expansion>
134 <char name='Source' vname='_dta'>http://support.sas.com/event
135 <char name='Origin' vname='_dta'>http://support.sas.com/event
136 <char name='universe' vname='chair'>Heroic, public spirited i
137 <char name='universe' vname='city'>City with adequate facilit
138 <char name='universe' vname='state'>U.S. State or Canadian Pr
139 <char name='_lang_list' vname='_dta'>default</char>
140 <char name='_lang_c' vname='_dta'>default</char>
141 <char name='note1' vname='_dta'>&quot;From 1976 Through 1984
142 <char name='note0' vname='_dta'>1</char>
143 <char name='note1' vname='OnLexJansen'>Indicates whether page
144 <char name='note0' vname='OnLexJansen'>1</char>
145 </expansion>
```

Variable labels

Other metadata

Our XML Example - 5

```
146 <data>
147 <o>
148 <v>SAS Global Forum 2009</v>
149 <v>http://support.sas.com/events/sasglobalforum/2009/index.ht
150 <v>Gaylord National Resort and Convention Center</v>
151 <v>Washington</v>
152 <v>DC</v>
153 <v>Lori Griffin</v>
154 <v>3328</v>
155 <v>http://support.sas.com/resources/papers/proceedings09/TOC.
156 <v>17978.0000000000000000</v>
157 <v>17981.0000000000000000</v>
158 <v>2</v>
159 </o>
160 <o>
161 <v>SAS Global Forum 2008</v>
162 <v>http://support.sas.com/events/sasglobalforum/2008/index.ht
163 <v>Henry B. Gonzalez Convention Center</v>
164 <v>San Antonio</v>
165 <v>Texas</v>
166 <v>Warren E. Stinson</v>
167 <v>3794</v>
168 <v>http://www2.sas.com/proceedings/forum2008/TOC.html</v>
169 <v>17607.0000000000000000</v>
170 <v>17610.0000000000000000</v>
171 <v>2</v>
172 </o>
```

<o> contains a row

<v> contains a column within a row (5th column contains "Texas")

Our XML Example - 6

```
589 </data>
590 <value_labels>
591 <vallab name='OnLexJansen'>
592 <label value='0'>No Papers Available</label>
593 <label value='1'>Some Papers Available</label>
594 <label value='2'>All Papers Available</label>
595 </vallab>
596 <vallab name='OnJansen'>
597 <label value='0'>None</label>
598 <label value='1'>Partial</label>
599 <label value='2'>Complete</label>
600 </vallab>
601 </value_labels>
602 </dta>
603
```

</data> ends the data element

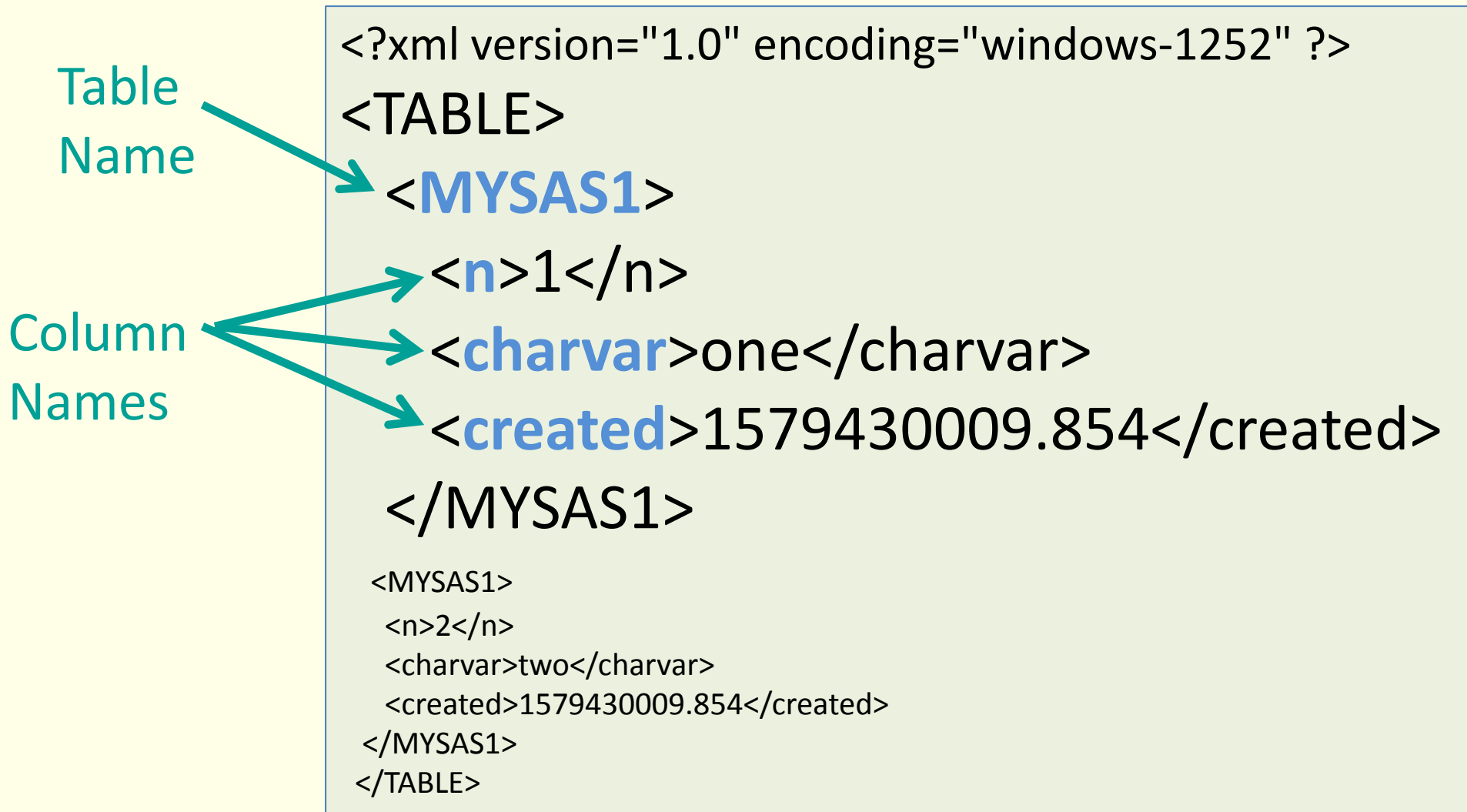
Value labels –
Something like
SAS formats
except for
numerics only

XML for Data – Two Approaches

- “Ad Hoc”
 - Element names are metadata
- Fixed Structure
 - Predefined element names
 - All data and metadata in content, not structure

“Ad Hoc – element names are also content”

- Example – Default SAS[®] Libname Engine XML



“Ad Hoc”

- Advantages
 - Human readable
 - Relatively Compact
 - Simple
- Disadvantages
 - Every table has a different set of element names
 - Where do metadata go?

Fixed Element and Attribute Names

- Example SAS “xmltype=export”
 - Structure has fixed element names
 - Variable names as values of attributes

<DATA>
defines a row

```
<TABLE-DATA>
```

```
<DATA>
```

```
<DATUM-NUMERIC name="n">10</DATUM-NUMERIC>
```

```
<DATUM name="charvar">ten</DATUM>
```

```
<DATUM-NUMERIC name="created">
```

```
1579430010.197
```

```
</DATUM-NUMERIC>
```

```
</DATA>
```

```
<DATA>
```

```
<DATUM-NUMERIC name="n">11</DATUM-NUMERIC>
```

```
<DATUM name="charvar">eleven</DATUM>
```

```
<DATUM-NUMERIC name="created">1579430010.197</DATUM-NUMERIC>
```

```
</DATA>
```

```
</TABLE-DATA>
```

Fixed Element and Attribute Names

- Example SAS “xmltype=export”
 - Structure has fixed element names
 - Variable names as values of attributes

<DATAUM-NUMERIC>
defines a column



```
<TABLE-DATA>
```

```
<DATA>
```

```
<DATUM-NUMERIC name="n">10</DATUM-NUMERIC>
```

```
<DATUM name="charvar">ten</DATUM>
```

```
<DATUM-NUMERIC name="created">
```

```
1579430010.197
```

```
</DATUM-NUMERIC>
```

```
</DATA>
```

```
<DATA>
```

```
<DATUM-NUMERIC name="n">11</DATUM-NUMERIC>
```

```
<DATUM name="charvar">eleven</DATUM>
```

```
<DATUM-NUMERIC name="created">1579430010.197</DATUM-NUMERIC>
```

```
</DATA>
```

```
</TABLE-DATA>
```

Fixed Element and Attribute Names

- A place for metadata

```
<TABLE name="mySAS2">
  <TABLE-HEADER>
    <Provider>SAS Institute Inc.</Provider>
  ...
  <TABLE-METADATA>
  ...
    <COLUMN order="3" name="created">
      <TYPE>numeric</TYPE>
      <DATATYPE>float</DATATYPE>
      <FORMAT>B8601DT</FORMAT>
    </TABLE-METADATA>
```

Fixed Element and Attribute Names

- Disadvantage
 - Can be more verbose
 - Less human readable ?
- Advantage
 - More machine actionable
 - Structured metadata with the data
 - In our case – we can make a SAS XML Map

Hierarchy, One to Many

One value_labels

<value_labels>

<vallab name='SeasonFR'>

```
<label value='1'>Hiver</label>
<label value='2'>Printemps</label>
<label value='3'>Ete</label>
<label value='4'>Automne</label>
</vallab>
```

<vallab name='Season'>

```
<label value='1'>Winter</label>
<label value='2'>Spring</label>
<label value='3'>Summer</label>
<label value='4'>Fall</label>
</vallab>
```

<vallab name='OnJansen'>

```
<label value='0'>None</label>
<label value='1'>Partial</label>
<label value='2'>Complete</label>
</vallab>
```

</value_labels>

Many vallab
(variable number)

XMLMap – Hierarchy, One to Many

For each vallab

```
<value_labels>
```

```
<vallab name='SeasonFR'>
```

```
<label value='1'>Hiver</label>
```

```
<label value='2'>Printemps</label>
```

```
<label value='3'>Ete</label>
```

```
<label value='4'>Automne</label>
```

```
</vallab>
```

```
<vallab name='Season'>
```

```
<label value='1'>Winter</label>
```

```
<label value='2'>Spring</label>
```

```
<label value='3'>Summer</label>
```

```
<label value='4'>Fall</label>
```

```
</vallab>
```

```
<vallab name='OnJansen'>
```

```
<label value='0'>None</label>
```

```
<label value='1'>Partial</label>
```

```
<label value='2'>Complete</label>
```

```
</vallab>
```

```
</value_labels>
```

Many label elements
(variable number)

XMLMap – Hierarchy to Relational Tables

<value_labels>

<vallab name='SeasonFR'>

<label value='1'>Hiver</label>

<label value='2'>Printemps</label>

<label value='3'>Ete</label>

<label value='4'>Automne</label>

</vallab>

<vallab name='Season'>

<label value='1'>Winter</label>

<label value='2'>Spring</label>

<label value='3'>Summer</label>

<label value='4'>Fall</label>

</vallab>

<vallab name='OnJansen'>

<label value='0'>None</label>

<label value='1'>Partial</label>

<label value='2'>Complete</label>

</vallab>

</value_labels>

vallab table

	value_labels_ORDINAL	vallab_ORDINAL	name
1	1	1	SeasonFR
2		1	Season
3		1	OnJansen

label table

	vallab_ORDINAL	label_ORDINAL	value	label
1	1	1	1	Hiver
2		1	2	Printemps
3		1	3	Ete
4		1	4	Automne
5		2	5	1 Winter
6		2	6	2 Spring
7		2	7	3 Summer
8		2	8	4 Fall
9		3	9	0 None
10		3	10	1 Partial
11		3	11	2 Complete

XMLMap File – From Hierarchy to Tables


FROM

```
<value_labels>
<vallab name='SeasonFR'>
...

```

XMLMap

TO



	value_labels_ORDINAL	vallab_ORDINAL	name
1	1	1	SeasonFR
2		1	Season
3		1	OnJansen

```
<TABLE name="vallab">
  <TABLE-DESCRIPTION>vallab</TABLE-DESCRIPTION>
  <TABLE-PATH syntax="XPath">/dta/value_labels/vallab</TABLE-PATH>
  ...
  <COLUMN name="name">
    <PATH syntax="XPath">/dta/value_labels/vallab/@name</PATH>
    <TYPE>character</TYPE>
    <DATATYPE>string</DATATYPE>
    <LENGTH>8</LENGTH>
  </COLUMN>
</TABLE>
```

XMLMap File – Rows

FROM

TO

```
<value_labels>  
<vallab name='SeasonFR'>  
...  
</vallab>  
</value_labels>
```

	value_labels_ORDINAL	vallab_ORDINAL	name
1	1	1	SeasonFR
2		1	Season
3		1	OnJansen

What Element Denotes ROWS

XMLMap

```
<TABLE name="vallab">  
<TABLE-DESCRIPTION>vallab</TABLE-DESCRIPTION>  
<TABLE-PATH syntax="XPath">/dta/value_labels/vallab</TABLE-PATH>
```

Described by an "XPATH"

XMLMap File – Columns

FROM

TO

```
<value_labels>  
<vallab name='SeasonFR'>  
...  
</vallab>  
</value_labels>
```

	value_labels_ORDINAL	vallab_ORDINAL	name
1	1	1	SeasonFR
2		1	Season
3		1	OnJansen

Which Elements/Attributes Denote Columns

XMLMap

```
<COLUMN name="name">  
<PATH syntax="XPath">/dta/value_labels/vallab/@name</PATH>  
<TYPE>character</TYPE>  
<DATATYPE>string</DATATYPE>  
<LENGTH>8</LENGTH>  
</COLUMN>
```


Also described by an XPATH

XMLMap File – Rows Numbered

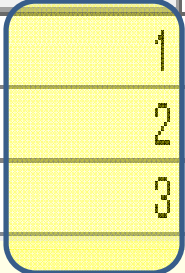
FROM

```
<value_labels>  
<vallab name='SeasonFR'>  
...  
</vallab>  
</value_labels>
```

TO



	value_labels_ORDINAL	vallab_ORDINAL	name
1	1	1	SeasonFR
2		1	Season
3		1	OnJansen



**Ordinals can be generated
to uniquely identify rows**

XMLMap File – Retained Information

FROM

TO

```
<value_labels>  
<vallab name='SeasonFR'>  
...  
</vallab>  
</value_labels>
```



	value_labels_ORDINAL	vallab_ORDINAL	name
1		1	SeasonFR
2		1	Season
3		1	OnJansen

**Data higher in the hierarchy
can be retained as the XML
is parsed**

XML Mapper – GUI for Making XMLMap Files

The screenshot displays the SAS XML Mapper interface. The left pane shows a tree view of the XML schema, with the 'vallab' element selected. The right pane shows the configuration for the 'vallab' element, where the 'Ordinal' option is selected. The 'Increment path' is set to '/dta/value_labels'. Below the configuration, a table view shows the resulting XML map for the 'vallab' element.

value_labels_ORDINAL	vallab_ORDINAL	name
1	1	SeasonFR
2	1	Season
3	1	OnJansen

Auto generated map

XML Mapper – Drag and Drop From Structure

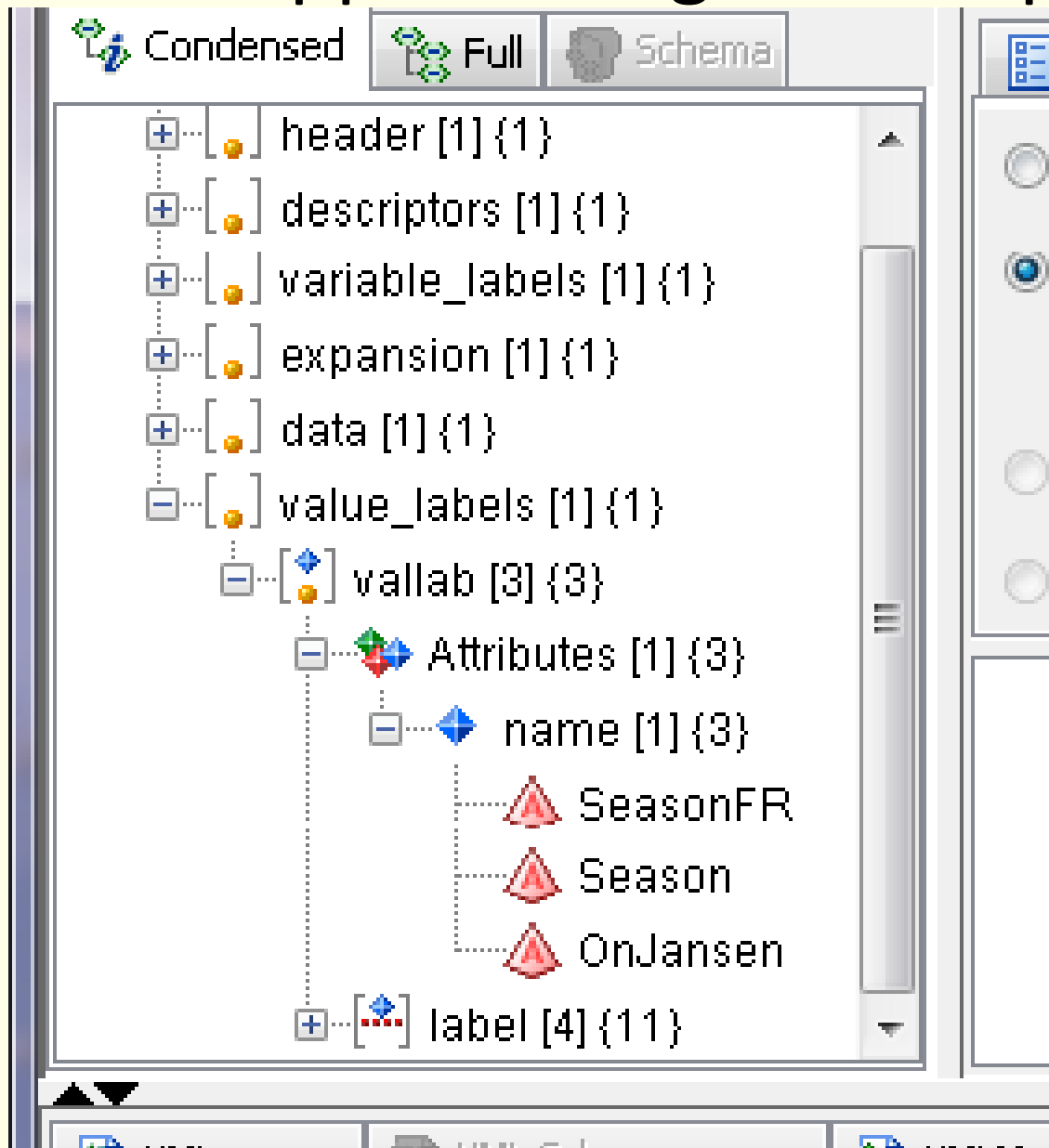


Table Definition (Column vallab_ORDINAL)

The screenshot displays a software interface for defining a table. The top section, titled 'Properties', contains the following fields and options:

- Name:** vallab_ORDINAL
- Description:** (empty field)
- Path:** (empty field)
- End Path:** (empty field)
- Retain:**
- Replace:**

The bottom section shows a tree view of the project structure:

- value_labels
- vallab
 - value_labels_ORDINAL
 - vallab_ORDINAL** (highlighted with a blue box)
 - name

A green arrow points from the text box to the 'vallab_ORDINAL' element in the tree view.

Row element
defines table

Table Definition (Column vallab_ORDINAL)

The screenshot displays a software interface for defining a table column. The top section, titled "Properties", shows the following fields:

- Name: vallab_ORDINAL
- Description: (empty)
- Path: (empty)
- End Path: (empty)

Below the properties section, there are two checkboxes: Retain and Replace.

The bottom section shows a tree view of the table structure:

- value_labels (expanded)
- vallab (expanded)
 - value_labels_ORDINAL
 - vallab_ORDINAL (highlighted with a blue box)
 - name

On the right side, a text box contains the text "Column elements within tables". Three teal arrows point from this text box to the "value_labels_ORDINAL", "vallab_ORDINAL", and "name" entries in the tree view.

Ordinals

The screenshot shows the 'Ordinals' dialog box in STATA. The 'Ordinal' radio button is selected. The 'Increment path' field is set to '/dta/value_labels/vallab'. A teal arrow points from a text box on the right to this field. Below the dialog is a tree view showing the following structure:

- value_labels
 - vallab
 - value_labels_ORDINAL
 - vallab_ORDINAL**
 - name

Element on
which to
increment

What if Order Matters?

<data>

<o>

<v>SUGI '76</v>

<v></v>

<v>Hyatt World</v>

<v>Kissimmee</v>

</o>

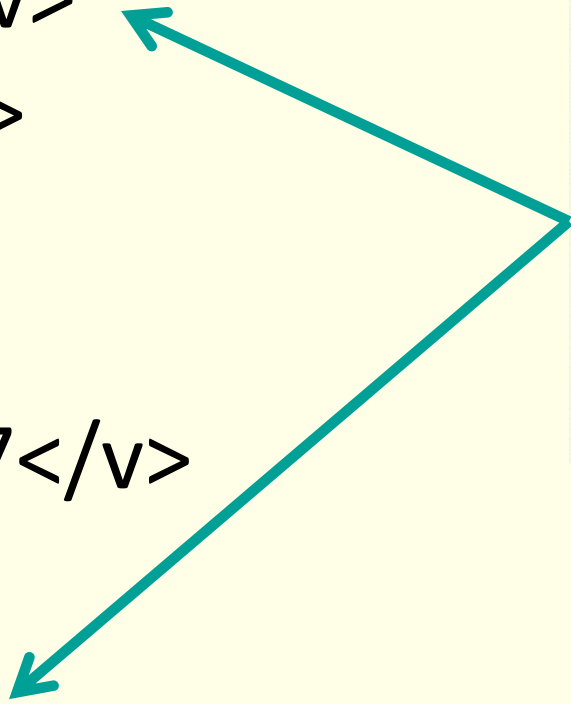
<o>

<v>SUGI '77</v>

<v></v>

<v>Fairmont</v>

Third <V>
within an
<o> is
column 3



Positional – using the order of elements


Third <V>
within an
<o> is
column 3

Table: o Row: 1 / 34 Columns: 1 / 4

	o	v1	v2	v3
1	1	SUGI'76		Hyatt World
2	2	SUGI'77		Fairmont
3	3	SUGI'78		Caesar's Palace

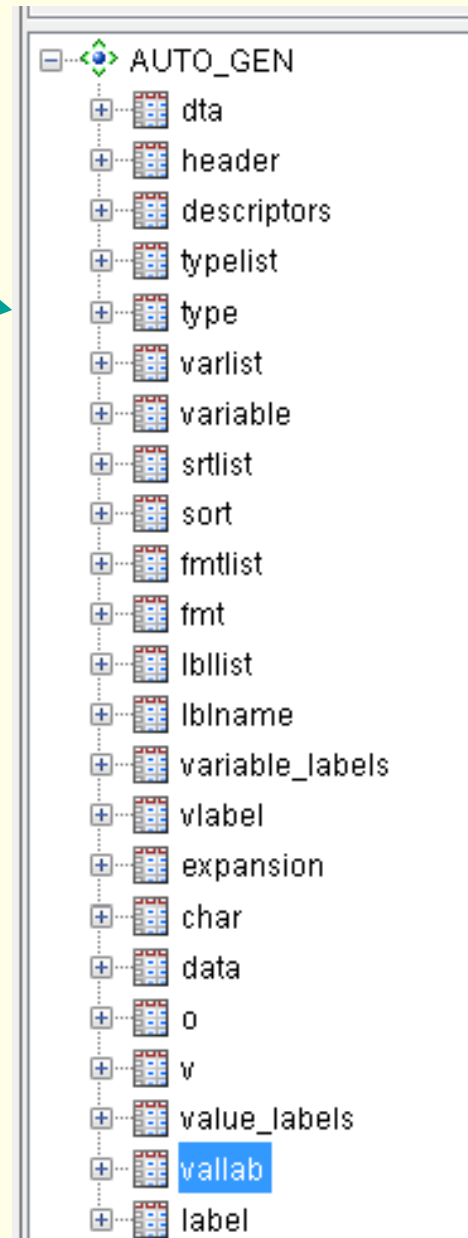
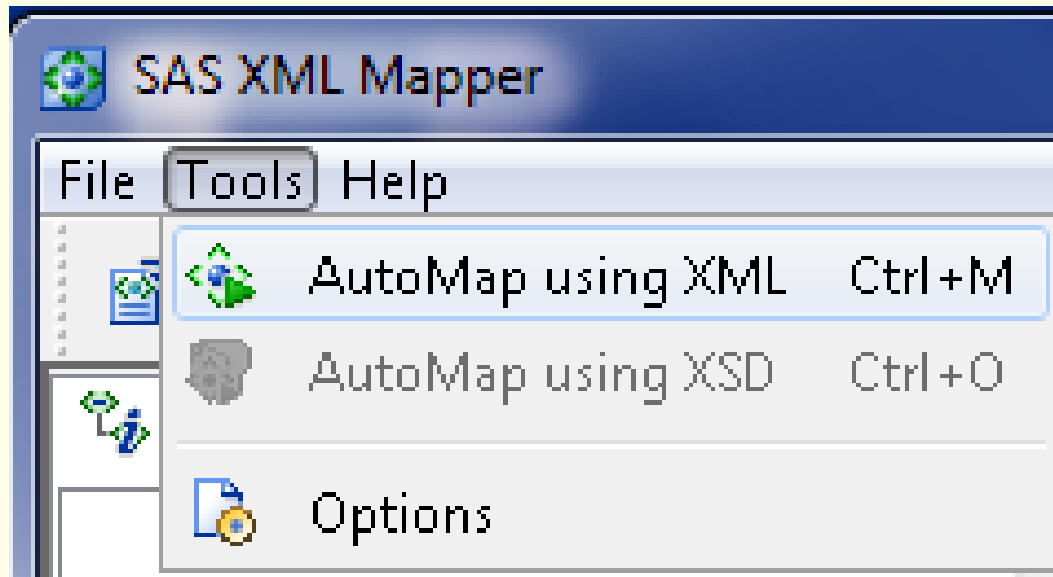
XML Mapper – Table Preview

Table: vallab Row: 1 / 3 Columns: 1 / 3

 SAS formats and informats are

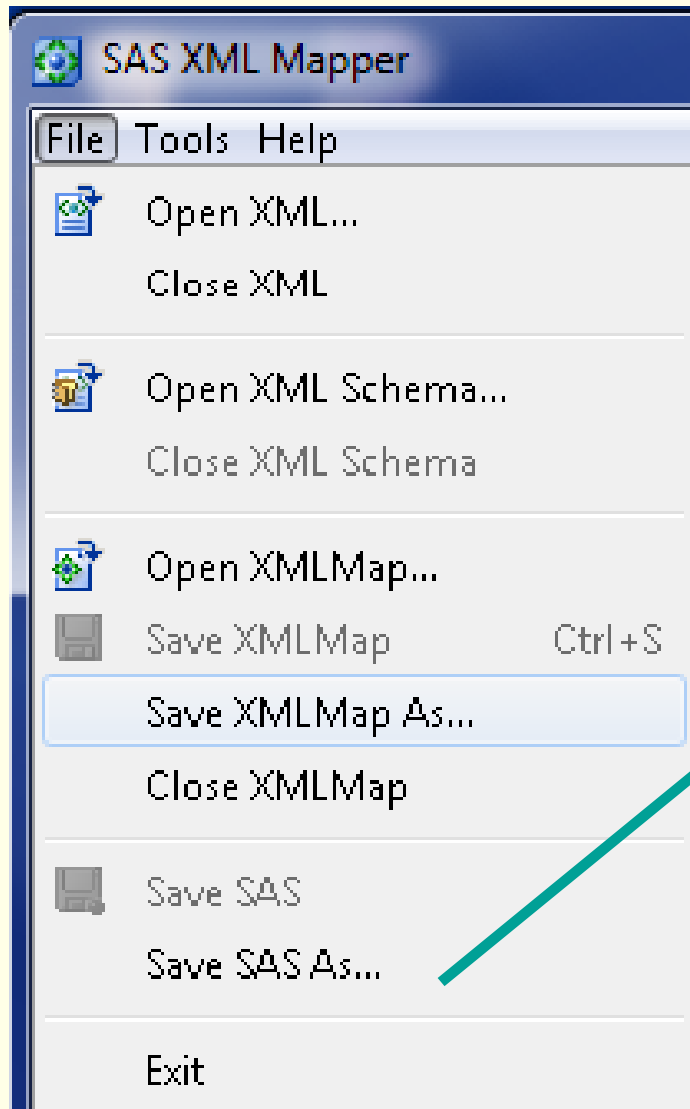
	value_labels_ORDINAL	vallab_ORDINAL	name
1	1	1	SeasonFR
2	1	2	Season
3	1	3	OnJansen

Handy Feature - Automap



For Our File:
One Click
Makes All
These Tables

Save XMLMap File and SAS Code



```
/* *****  
 * Generated by XML Mapper,  
 ***** */  
/* * Environment */  
filename SUGISGF3 'C:\SUGI_SGF3.xml';  
filename SXLEMAP  
'C:\SGF_030_2010_XMLmap.map';  
libname SUGISGF3 xml xmlmap=SXLEMAP  
access=READONLY;  
  
/* * Catalog */  
proc datasets lib=SUGISGF3; run;  
  
/* * Contents */  
proc contents data=SUGISGF3.dta varnum; run;
```

SAS Code

```
/* *****  
 * Generated by XML Mapper,  
 ***** */  
/* * Environment */  
filename SUGISGF3 'C:\SUGI_SGF3.xml';  
filename SXLEMAP 'C:\SGF_030_2010_XMLmap.map';  
  
libname SUGISGF3 xml xmlmap=SXLEMAP  
       access=READONLY;  
  
/* * Catalog */  
proc datasets lib=SUGISGF3; run;  
  
/* * Contents */  
proc contents data=SUGISGF3.dta varnum; run;
```

This XML



Mapped by
this XMLMap
file



SGF 2010 Paper

Paper 157-2010 Lex Jansen

*Understanding the define.xml File and
Converting It to a Relational Database*

Prototype or Schema?

- Prototype may be easier to understand
- Schema covers all possibilities, prototype may not use some features
 - Maybe what you want, maybe not
- Schema may be unusable
 - Possibly too large
 - Recursion can make for infinite depth

XML Mapper Issues with “Real World” XML

- Prototype not fully representative
- “Invalid” XML
 - XML does not match DTD or schema
 - Lower validation level in XML Mapper?

```
Attribute value "" of type IDREF must be an NCName when namespaces are enabled.
```

- SAS libname engine does not process DTDs
 - Ignore?

“WARNING: DOCTYPE element encountered. The SAS XML Libname Engine does not support processing of Data Type Definitions (DTD). External entity references in the document will not be resolved, and no mark up validation will be performed.”

Everything is in Tables: All Done?

Not Yet

From 23 tables to 3 tables

- XML Map generates 23 tables
- We need 3
 - Dataset
 - CNTLIN dataset
 - Other metadata dataset

Content / Role	Table(s)
Data - columns	v
Data - rows	o
List of variables	variable
Variable labels	vlabel
Data type and length	fmt, type
Dataset information - label, timestamp, nobs etc	header, char
Sort order	sort
Value labels (formats)	vallab, label
Link formats to variables	lblname
Other variable characteristics	char

Stata “dta” XML file TO SAS Dataset Tasks

- Data
 - Transpose from one number per row to one row per observation
 - Name columns
 - Convert from text to proper type (e.g. numeric, dates)
- Assign dataset, and variable labels
- (value) “labels” to SAS formats or “formats” to formats
- Capture other metadata
 - Notes
 - Characteristics

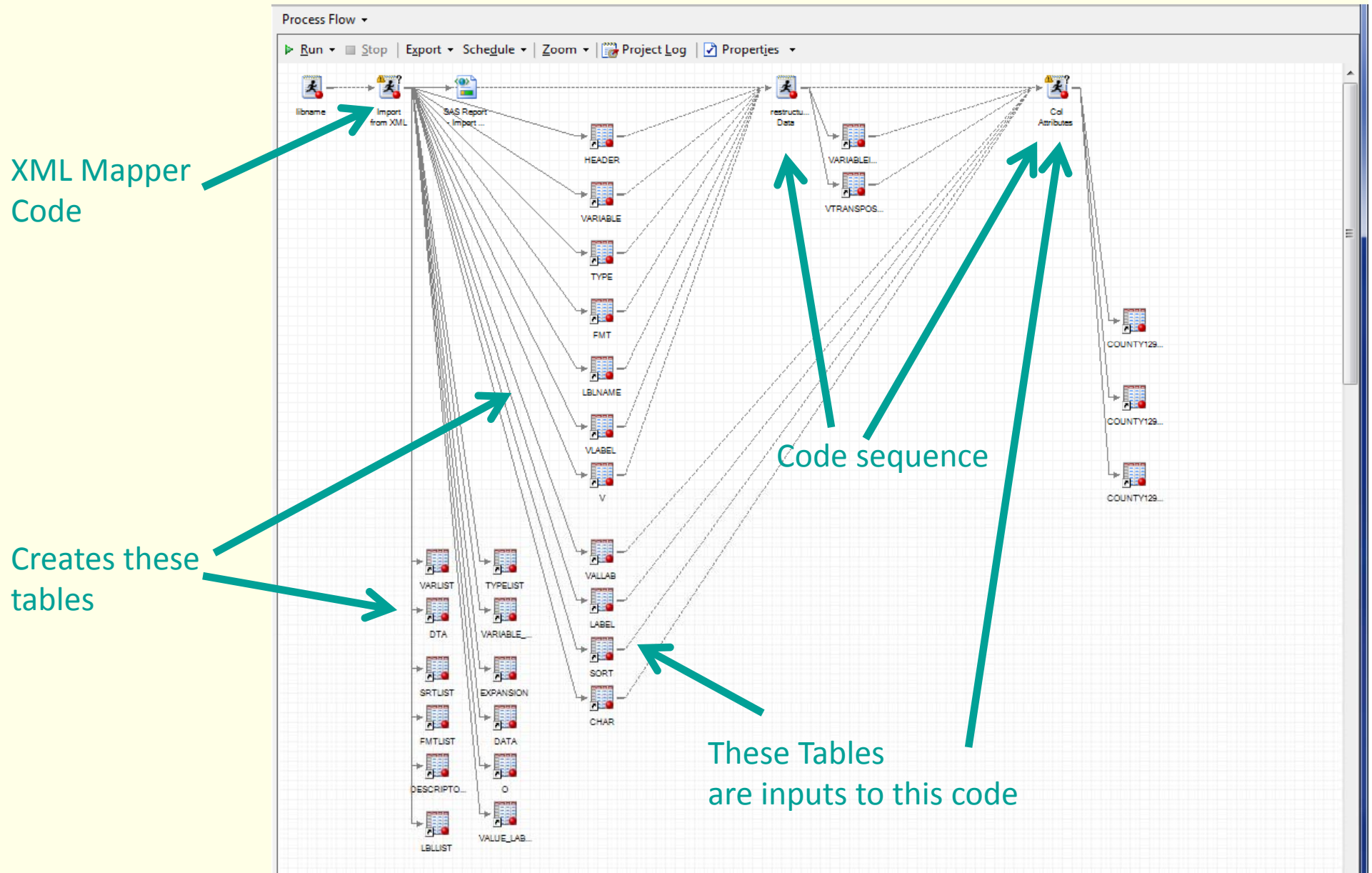
Build a Repeatable Process

- Create a process that converts ANY XML file of this type
- Simple to use
- Documented

Document

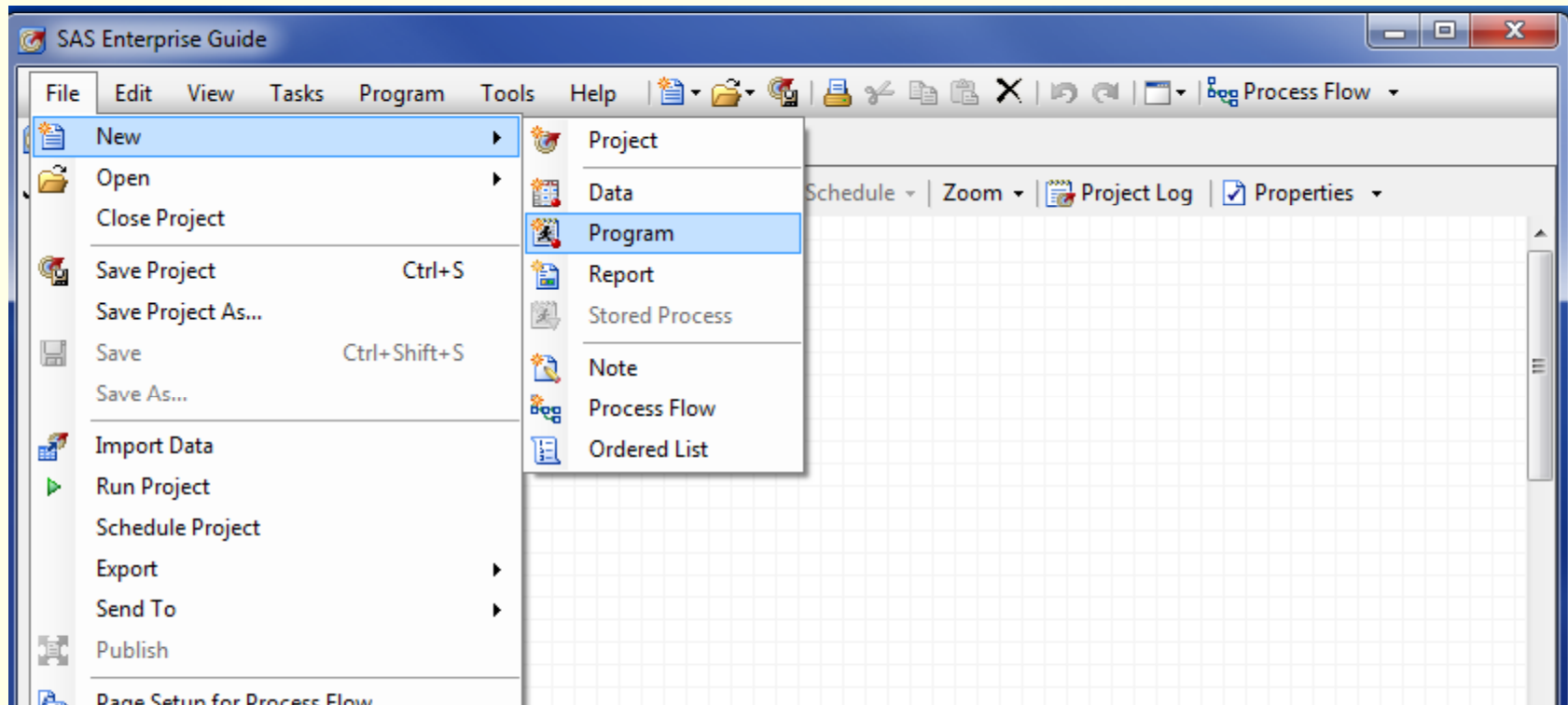
- Sequence of steps
- Which tables are used for what
- Enterprise Guide? (EG)

EG Process Flow Diagram

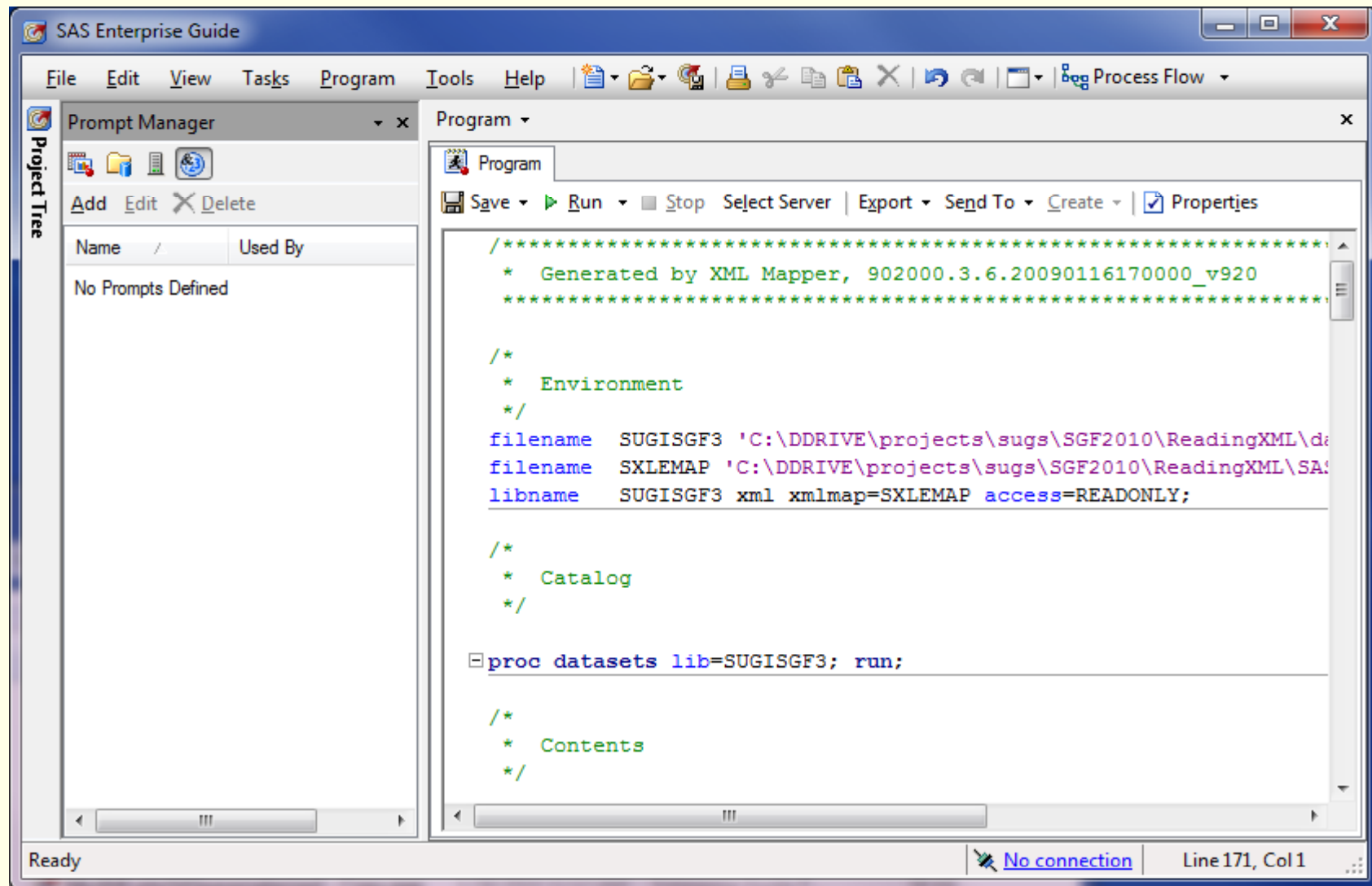


Enterprise Guide Project SAS Code from XML Mapper

- We could link to external code but
 - Copying puts it in the project
 - We'll modify the code



Paste the Code from XML Mapper



Run the Code

The screenshot shows the SAS Enterprise Guide interface. The main window displays the results of a program execution, showing a directory listing and a table of member types.

Directory Listing:

Directory	
Libref	SUGISGF3
Engine	XML
Access	READONLY
Physical Name	SUGISGF3
XMLType	GENERIC
XMLMap	SXLEMAP

Member Type Table:

#	Name	Member Type
1	CHAR	DATA
2	DATA	DATA
3	DESCRIPTORS	DATA
4	DTA	DATA
5	EXPANSION	DATA
6	FMT	DATA
7	FMTLIST	DATA
8	HEADER	DATA
9	LABEL	DATA
10	LBLLIST	DATA
11	LBLNAME	DATA
12		DATA

The interface also shows a Project Tree on the left with 'No Prompts Defined' and a status bar at the bottom indicating 'Ready' and 'No connection'.

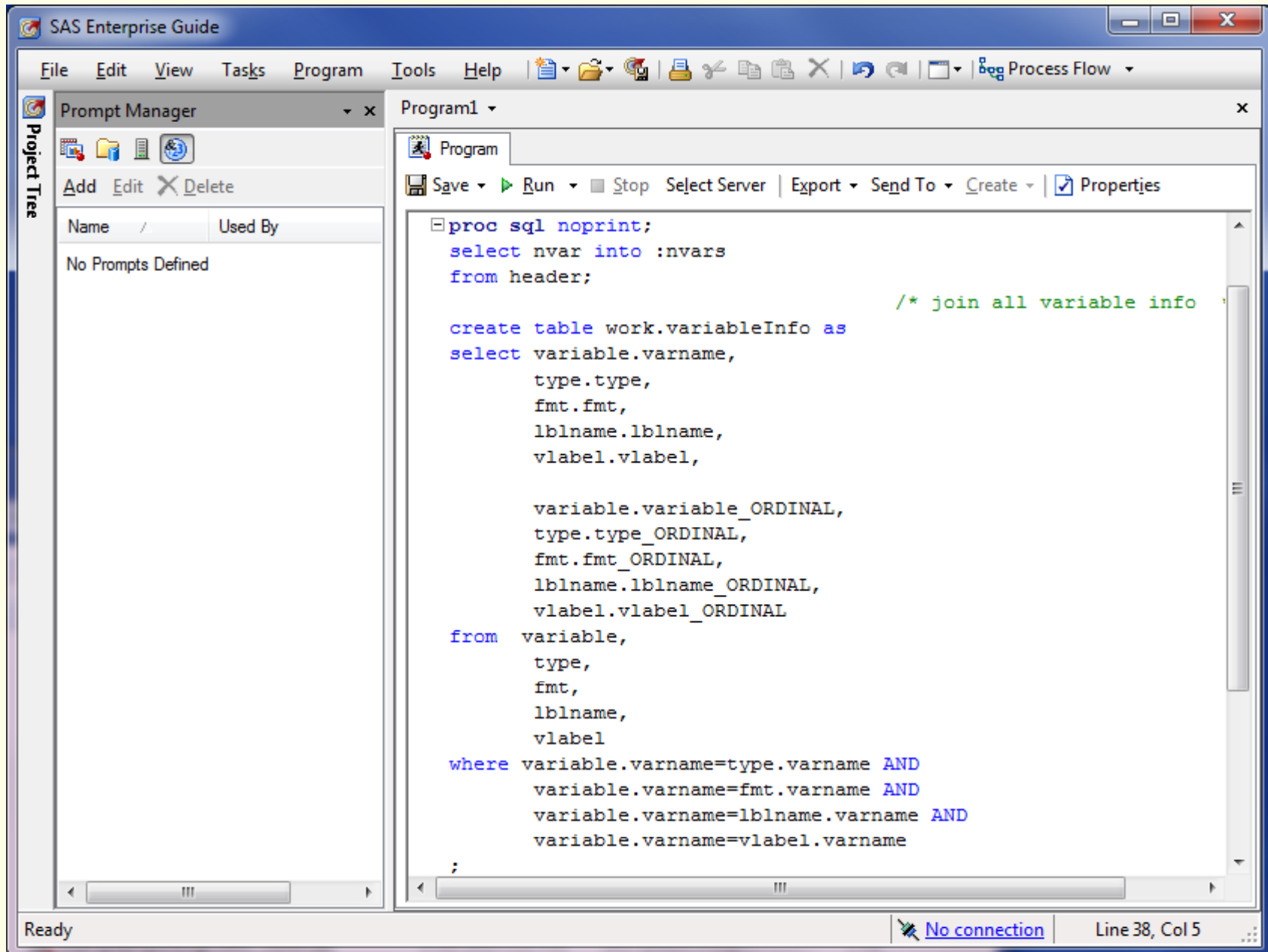
Process Flow

The screenshot displays the SAS Enterprise Guide interface. The main window is titled "Process Flow" and shows a diagram on a grid background. On the left, a "Project Tree" pane is visible, containing a "Prompt Manager" section with a table that currently shows "No Prompts Defined".

The "Process Flow" diagram consists of a single "Program" node on the left, which is connected by arrows to a vertical column of five nodes on the right: "DTA", "HEADER", "DESCRIPTO...", "TYPELIST", and "TYPE". Each node is represented by a small icon with a calendar-like grid. The "Program" node is a larger icon with a person running. The "DTA" node is a calendar icon with a red dot. The other nodes are similar calendar icons.

At the bottom of the window, the status bar shows "Ready" on the left and "No connection" on the right.

New Program Node – Aggregate Variable Info



The screenshot displays the SAS Enterprise Guide interface. The main window, titled "Program1", contains a code editor with the following SQL code:

```
proc sql noprint;
  select nvar into :nvars
  from header;

  /* join all variable info */
  create table work.variableInfo as
  select variable.varname,
         type.type,
         fmt.fmt,
         lblname.lblname,
         vlabel.vlabel,

         variable.variable_ORDINAL,
         type.type_ORDINAL,
         fmt.fmt_ORDINAL,
         lblname.lblname_ORDINAL,
         vlabel.vlabel_ORDINAL
  from variable,
       type,
       fmt,
       lblname,
       vlabel
  where variable.varname=type.varname AND
        variable.varname=fmt.varname AND
        variable.varname=lblname.varname AND
        variable.varname=vlabel.varname
  ;
```

The interface also shows a "Prompt Manager" window on the left with "No Prompts Defined" and a status bar at the bottom indicating "Ready" and "No connection".

Code We Write

```
create table work.variableInfo as
select variable.varname, type.type,
       fmt.fmt, lblname.lblname,
       vlabel.vlabel, variable.variable_ORDINAL,
       type.type_ORDINAL,   fmt.fmt_ORDINAL,
       lblname.lblname_ORDINAL, vlabel.vlabel_ORDINAL
from variable,
     type,
     fmt,
     lblname,
     vlabel
where variable.varname=type.varname   AND
       variable.varname=fmt.varname   AND
       variable.varname=lblname.varname AND
       variable.varname=vlabel.varname;
```

**Five tables
have
metadata
about
variables**

Table V Has Tall Skinny Data - Transpose

V

o_ORDINAL	v_ORDINAL	v
1	1	SUGI '76
1	2	
1	3	Hyatt World
1	4	Kissimmee
1	5	Florida
1	6	Julian Horwich
1	7	206
1	8	
1	9	5869.00000000
1	10	5871.00000000
1	11	0
1	12	1
2	13	SUGI '77
2	14	
2	15	Fairmont

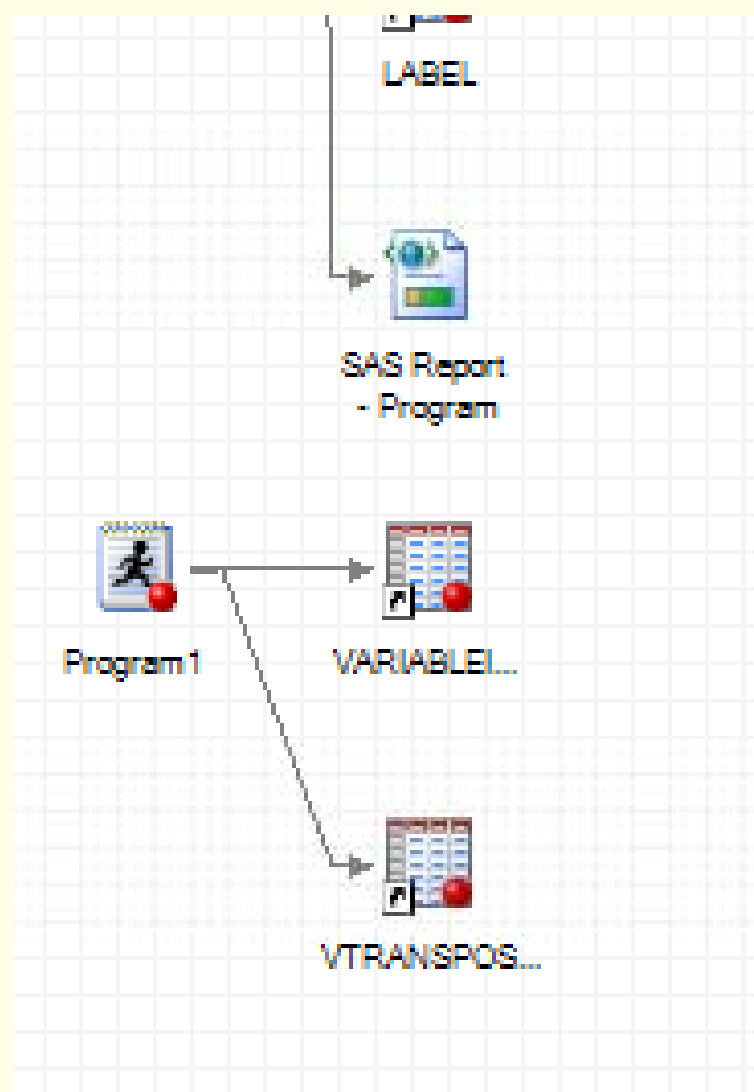
```
proc transpose
  data=work.v
  out=work.vTransposed;
  var v;
  by o_ordinal;
run;
```

VTransposed

o_ORDINAL	_NAME_	_LABEL_	COL1	COL2	COL3	COL4	COL5	COL6
1	v	v	SUGI '76		Hyatt World	Kissimmee	Florida	Julian Hor
2	v	v	SUGI '77		Fairmont	New Orleans	Louisiana	Rod Helms
3	v	v	SUGI '78		Caesar's Palace	Las Vegas	Nevada	Mike Farre
4	v	v	SUGI '79		Sheraton Sand K	Clearwater	Florida	Ramon Litt

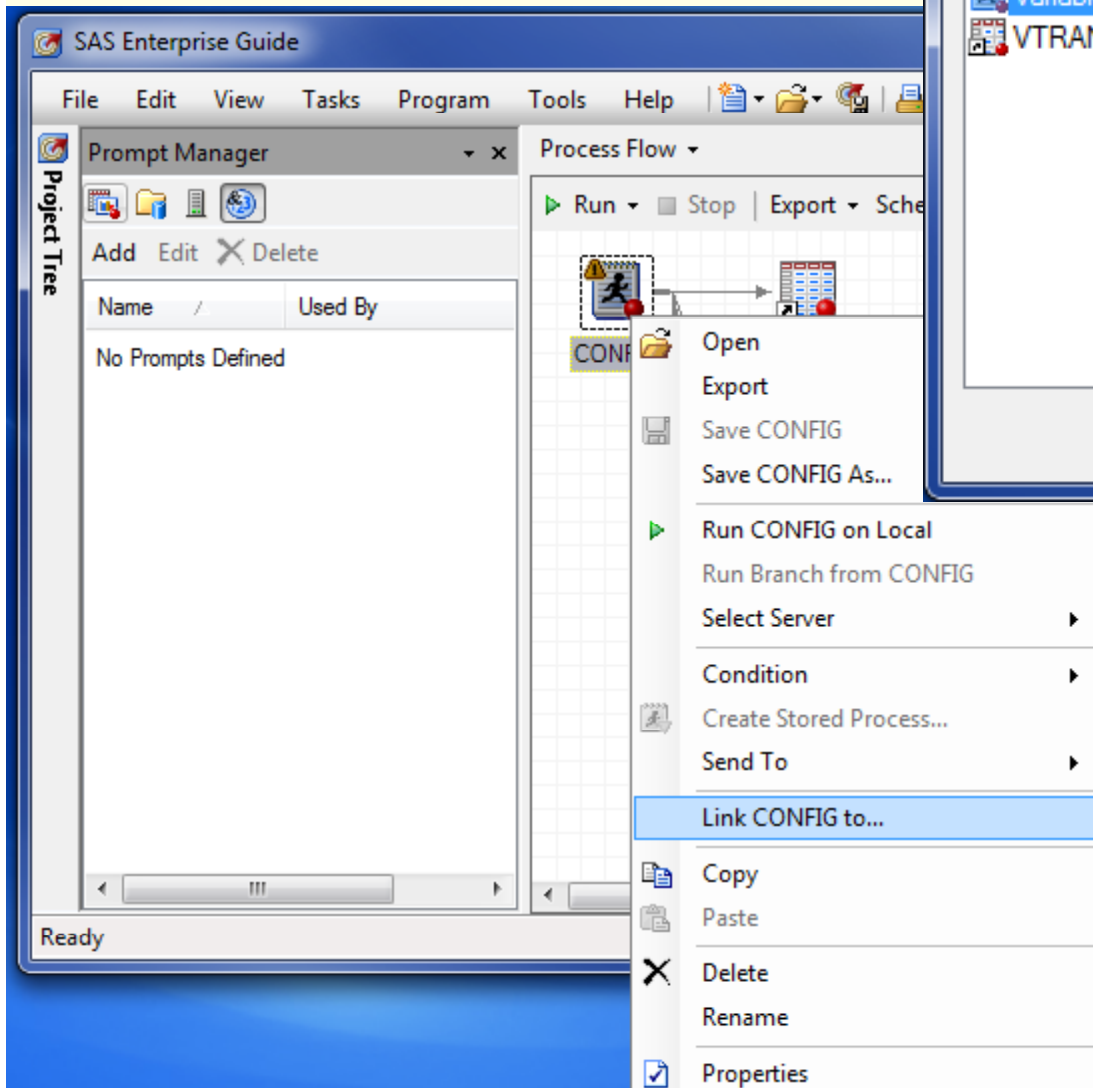
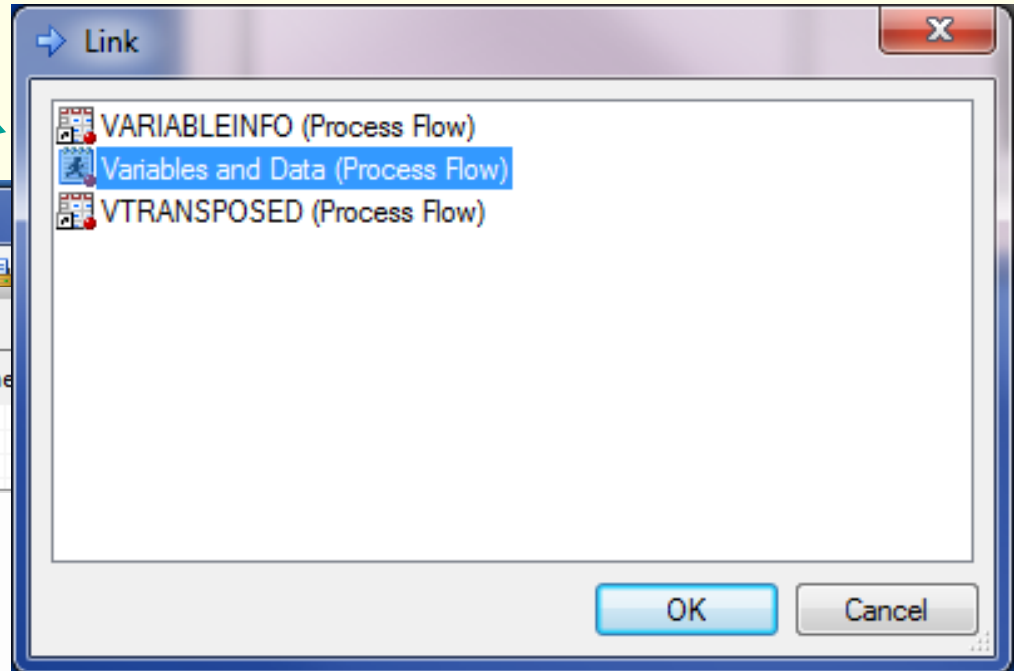
We're Building a Process

- Best to:
 - Give nodes useful names
 - Link steps and datasets



Linking the Program Nodes

Variables and Data

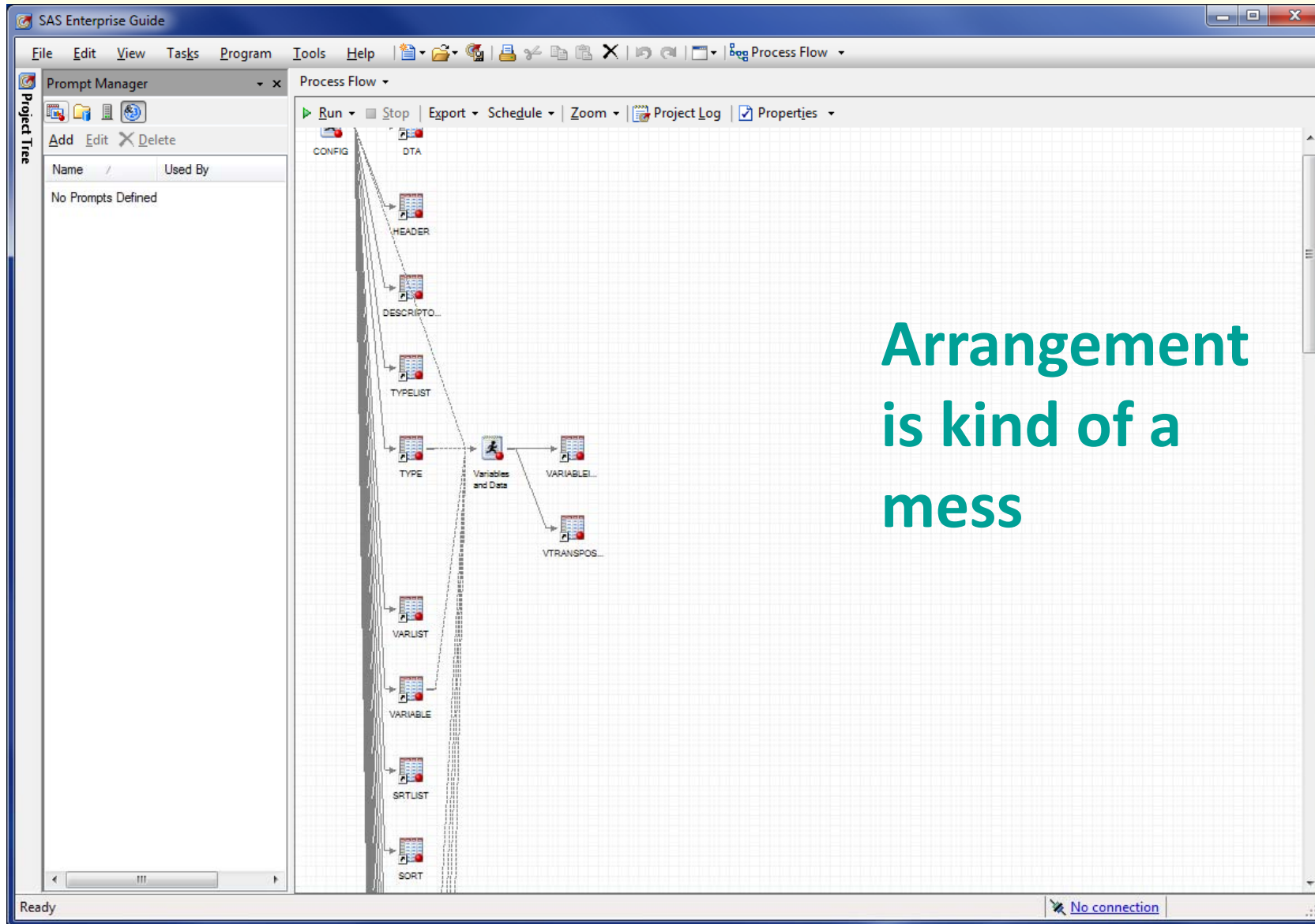


**Link
CONFIG
to...**

Linked

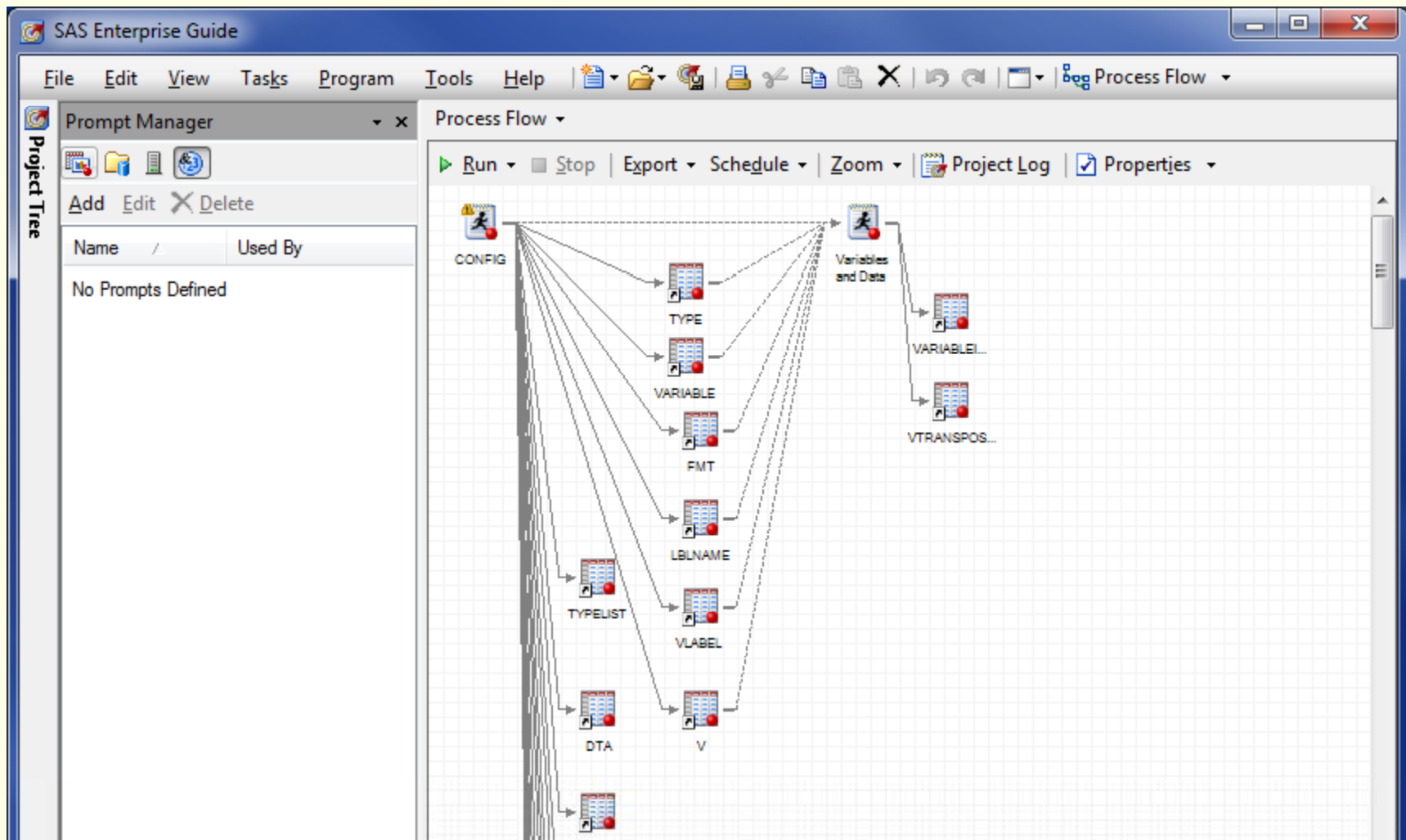
The screenshot displays the SAS Enterprise Guide interface. On the left is the 'Project Tree' pane with a 'Prompt Manager' sub-pane containing a table with columns 'Name' and 'Used By', and the text 'No Prompts Defined'. The main 'Process Flow' pane shows a workflow diagram on a grid. The workflow starts with a 'SAS Report Program' node, which is highlighted with a yellow oval. A teal arrow points from a yellow box labeled 'Link' to this node. Below the 'SAS Report Program' node are two other nodes: 'Variables and Data' and 'VTRANSPOS...'. Arrows indicate the flow from 'SAS Report Program' to both 'Variables and Data' and 'VTRANSPOS...'. The 'Variables and Data' node is further connected to a 'VARIABLEI...' node. The top menu bar includes 'File', 'Edit', 'View', 'Tasks', 'Program', 'Tools', and 'Help'. The status bar at the bottom shows 'Ready' and 'No connection'.

Also Link Contributing Datasets

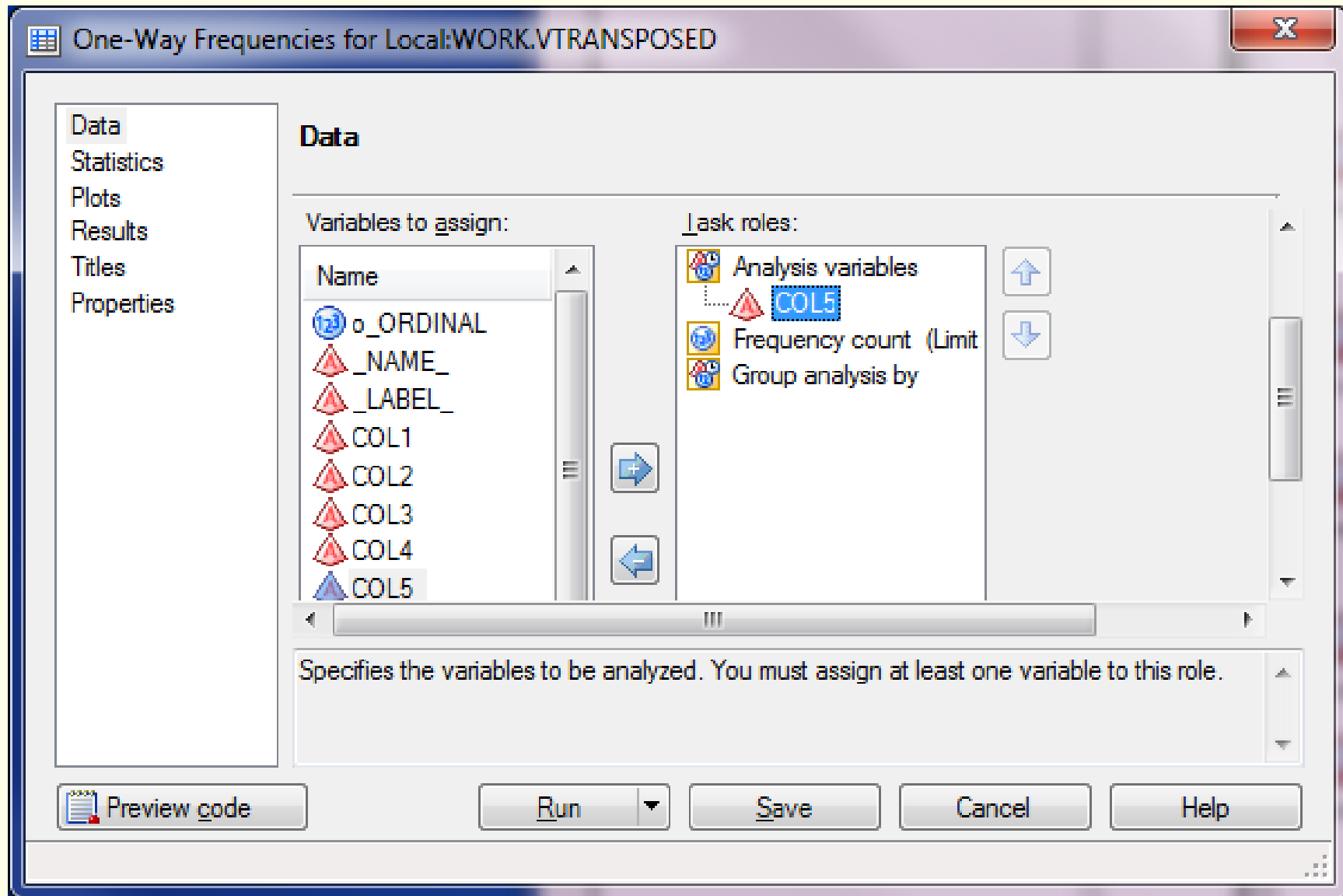


Arrangement
is kind of a
mess

Option – Turn Off AutoArrange – Layout Manually



Task Wizards Are Available



SUGI/SGF States




COL5	Frequency	Percent	Cumulative Frequency	Cumulative Percent
California	5	14.71	5	14.71
DC	1	2.94	6	17.65
Florida	9	26.47	15	44.12
Georgia	1	2.94	16	47.06
Hawaii	1	2.94	17	50.00
Illinois	1	2.94	18	52.94
Indiana	1	2.94	19	55.88
Louisiana	3	8.82	22	64.71
Nevada	2	5.88	24	70.59
New York	1	2.94	25	73.53
Pennsylvania	1	2.94	26	76.47
Quebec	1	2.94	27	79.41
Tennessee	2	5.88	29	85.29
Texas	4	11.76	33	97.06
Washington	1	2.94	34	100.00

CNTLIN Facility

Create Formats from a File

MYDATASET_CNTLIN ▾

 Filter and Sort  Query Builder | Data ▾ Describe ▾

	 fmtname	 start	 label
1	SeasonFR	1	Hiver
2	SeasonFR	2	Printemp
3	SeasonFR	3	Ete
4	SeasonFR	4	Automne
5	Season	1	Winter
6	Season	2	Spring
7	Season	3	Summer
8	Season	4	Fall
9	OnJansen	0	None
10	OnJansen	1	Partial
11	OnJansen	2	Complete

```
Proc format;
```

```
  Value Season
```

```
    1 = "Winter"
```

```
    2 = "Spring"
```

```
    3 = "Summer"
```

```
    4 = "Fall"
```

```
;
```

```
  Value OnJansen
```

```
    0 = "None"
```

```
    1 = "Parital"
```

```
    2 = "Complete"
```

```
;
```

Translating to SAS Formats

From	To
%21s	\$21.
%8.0g	Best8.
%td	Date.

Identify with Perl Regular Expressions

From

PRX

%21s

'/^%[-~]*(\d+)s/'

%8.0g

'/^%-*(\d+)\.(\d+)g/'

%td

'/^%td/'

Match the Type of Format

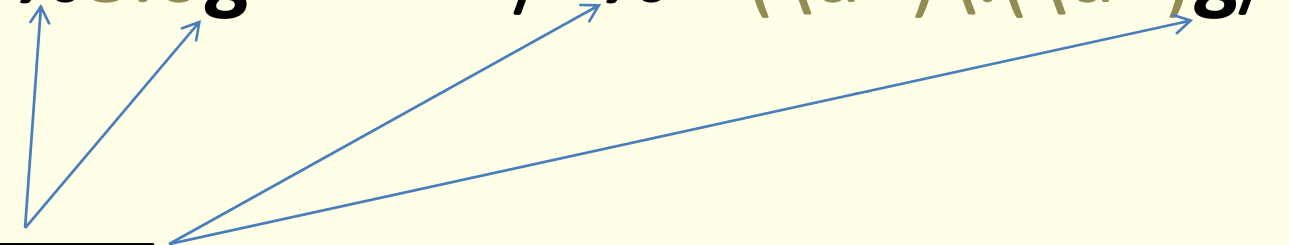
From

%8.0g

PRX

'/^%-*(\d+)\.(\d+)g/'

Type of
format



Split Out Pieces – One or More Digit Before the Decimal Point

From

`%8.0g`

PRX

`'/^%-*(\d+)\.(\d+)g/'`

Preceding
Decimal Point

String matched
within
parentheses
“captured”

Split Out Pieces – One or More Digit After the Decimal Point

From

`%8.0g`

PRX

`'/^%-*(\d+)\.(\d+)g/'`

Following Decimal
Point

String matched
within
parentheses
“captured”

An array of possibilities

30 Possible Patterns

```
array fmts {&nPatterns,3} $ 30
  _temporary_ (
    '/^%-* (\d+) \. (\d+) g/'      'BEST'      '2'
    '/^%-* (\d+) \. (\d+) f/'      ' '         '2'
    '/^%-* (\d+) \. (\d+) e/'      'E',       '2'
    '/^% (\d+) x/'                  'HEX'      '2'
```

**Base of SAS
Format**



**Number of
Numeric Pieces**



Parameterizing the Process with Prompts

Create the Prompt

Add New Prompt

General | Prompt Type and Values

Name:
XMLfile

Displayed text:
Enter the name of the dta XML file to read (example: MyData.xml)

Description:

Options

Hide from user Requires a non-blank value

Read-only values

Parameterizing the Process with Prompts

Create the Prompt

Name:

XMLfile

Displayed text:

Enter the name of the dta XML file to read (example: MyData.xml)

Displayed text:

Enter the name of the dta XML file to read (example: MyData.xml)

Description:

Options

Hide from user Requires a non-blank value

Read-only values

Parameterizing the Process with Prompts

Create the Prompt

Name:

XMLfile

Displayed text:

Enter the name of the dta XML file to read (example: MyData.xml)

**Macro
Variable**

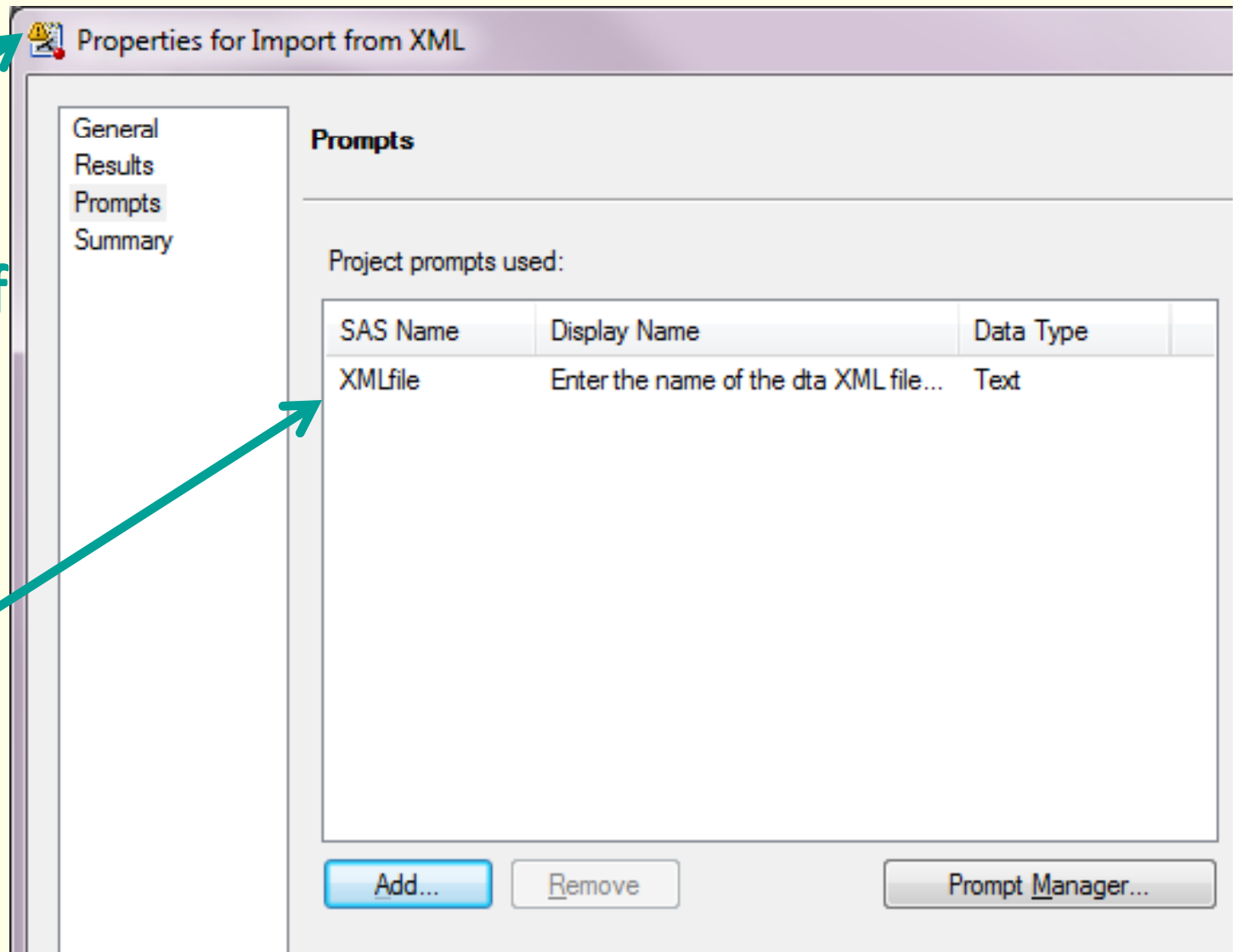
**Displayed
when the
user is
prompted**

Parameterizing the Process with Prompts

Associate it with a Code Node

Properties of a node

Prompt for the node



Parameterizing the Process with Prompts

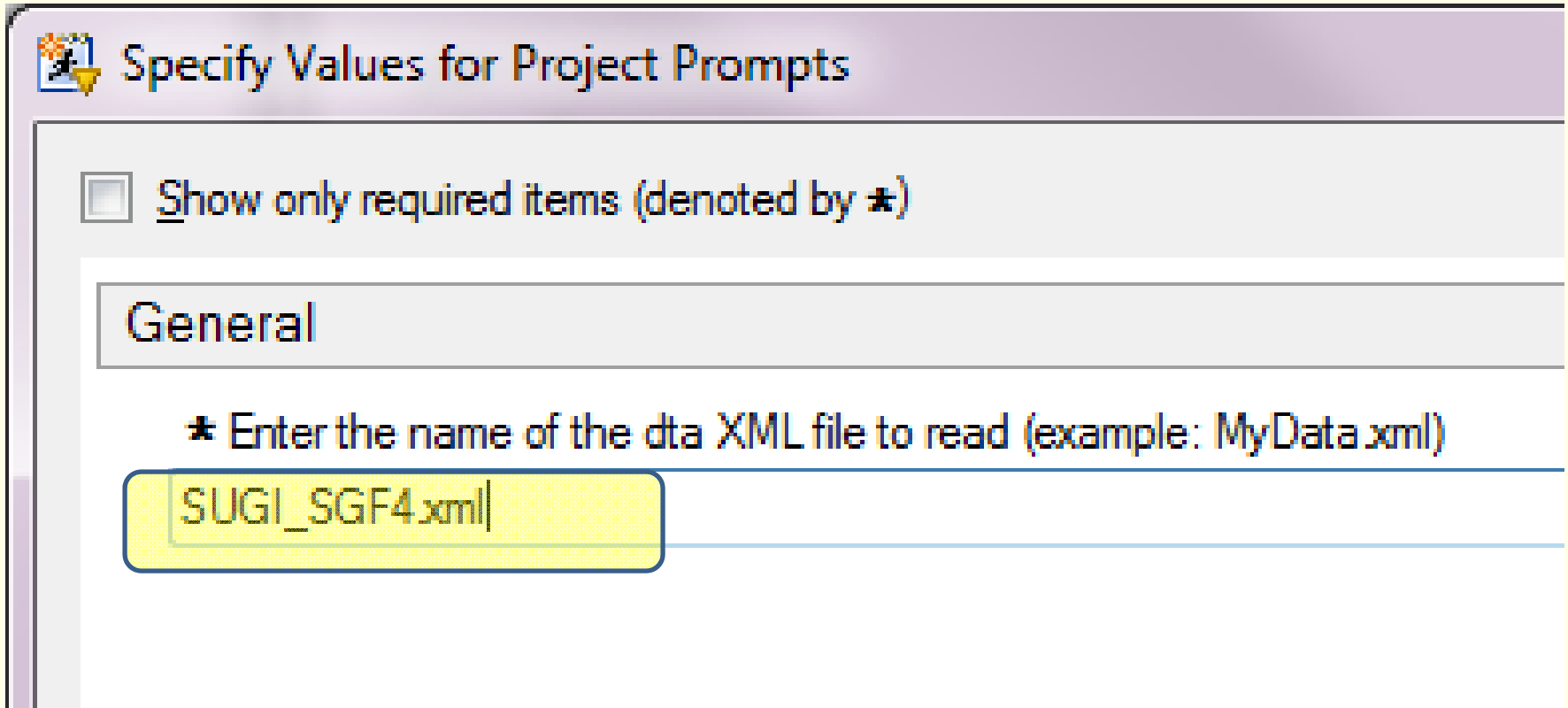
Use the Macro Variable in the Code

```
%let inFolder = C:\ ReadingXML\data\SUGI_SGF_files;  
%let mapFolder = C:\ReadingXML\SAScode;  
%let mapFile = SGF_030_2010_XMLmap.map;  
  
filename SUGISGF3 "&inFolder.\&XMLfile.";  
filename SXLEMAP "&mapFolder.\&mapFile.";  
libname SUGISGF3 xml xmlmap=SXLEMAP access=READONLY;
```



**Macro variable
reference**

When The Node Runs



The image shows a screenshot of a software dialog box titled "Specify Values for Project Prompts". The dialog has a purple header bar with a small icon on the left. Below the header, there is a checkbox labeled "Show only required items (denoted by *)". Underneath this is a section titled "General" in a light gray box. Below the "General" section, there is a text prompt: "* Enter the name of the dta XML file to read (example: MyData.xml)". Below the prompt is a text input field with a yellow background and a blue border, containing the text "SUGI_SGF4.xml".

Specify Values for Project Prompts

Show only required items (denoted by *)

General

* Enter the name of the dta XML file to read (example: MyData.xml)

SUGI_SGF4.xml

Other Metadata

object	object Label	Type	characteristic
_dta	Dataset: MyDataset	Source	http://support.sas.com/events/sasglobalforum/previous/index.html and http://www.lexjansen.com/sugi/
_dta	Dataset: MyDataset	Origin	http://support.sas.com/events/sasglobalforum/previous/index.html and http://www.lexjansen.com/sugi/
_dta	Dataset: MyDataset	note1	From 1976 Through 1984 the conferences were named SUGI 'YY where YY was the two digit year. From 1985 through 2006 the conferences were named SUGI nn, where nn was a sequential number beginning with 10. From 2007 through the present the conferences have been named SGF YYYY, where YYYY is the 4 digit year.
chair	Conference Chair(s)	universe	Heroic, public spirited individuals
city	Conference City	universe	City with adequate facilities for SAS international conference
OnLexJansen	Papers Available on lexjansen.com	note1	Indicates whether papers are available on Lex Jansen's SUGI paper site: http://www.lexjansen.com/sugi/
season	Season at Beginning of Conference	note1	can be labeled either by Season or SeasonFR
state	Conference State	universe	U.S. State or Canadian Province or Territory

Other Metadata

“Source” for the whole table (_dta)

object	object Label	characteristic Type	characteristic
_dta	Dataset: MyDataset	Source	http://support.sas.com/events/sasglobalforum/previous/index.html and http://www.lexjansen.com/sugi/

Other Metadata

“universe” for the variable “chair”

object	object Label	characteristic Type	characteristic
chair	Conference Chair(s)	universe	Heroic, public spirited individuals

Complete Code in the Paper

**[http://www.sascommunity.org/wiki/
Using_XML_Mapper_and_Enterprise_Guide_
to_Read_Data_and_Metadata_from_an_
XML_File](http://www.sascommunity.org/wiki/Using_XML_Mapper_and_Enterprise_Guide_to_Read_Data_and_Metadata_from_an_XML_File)**

SAScommunity.org

(search for “XML”) or find it in Sasopedia

Sasopedia

Sasopedia is a *place* where users of SAS can provide factual information and tips on using SAS software. There are various ways of organizing the information.

Categories are used to organize this information as discussed in [Organizing Articles](#) which is also found in the *getting started* box.

The **Sasopedia** is intended to serve the needs of what many people refer to as a FAQ (Frequently Asked Questions). Just click on the **[+]** to expand and see any subcategories. Just the link text to see both the articles and categories in that Sasopedia category.

Sasopedia is currently undergoing a re-org (e.g., making the category names consistent). If you are interested in helping with that please email [us](#).

Language Elements	Procedures	Products	Topics
[+] Formats	[+] APPEND Procedure	[+] SAS/ACCESS	[+] Analytics and Modeling
[+] Functions and CALL Routines	[+] CIMPORT Procedure	[+] SAS/AF	[+] Application Development
[+] Informat	[+] COMPARE Procedure	[x] SAS/ASSIST	[+] Automation
[+] Options	[+] CONTENTS Procedure	[+] SAS Add-In for Microsoft Office	[+] Best Practices
[+] Statements	[+] CONTENTS Procedure	[+] SAS9 BI/EBI	[-] Data Representation
	[+] CONTENTS Procedure	[+] Base SAS	[+] Dates and Times
	[+] CONTENTS Procedure	[+] SAS/CONNECT	[+] Missing Values
	[+] CONTENTS Procedure	[+] SAS/CONNECT	[+] XML

Enterprise Guide 4.3

<http://support.sas.com/documentation/cdl/en/whatsnew/62580/HTML/default/viewer.htm#egwhatsnew43.htm>

What's New in SAS Enterprise Guide 4.3

Overview

SAS Enterprise Guide 4.3 includes the following new features and enhancements:

- [new program editor with autocomplete and integrated syntax help](#)
- [recently accessed program files are available from the File menu](#)
- [option to automatically add output data to the project tree](#)
- [ability to analyze SAS programs](#)
- [explicit SQL pass-through option](#)
- [more control over whether ODS statements are generated for the results](#)
- [autoexec process flow](#)
- [ability to send data files to JMP](#)
- [longer names and custom filters for SAS Information Maps](#)
- [automatic chart option for OLAP cubes](#)
- [ability to create OLAP cube slices that use SAS code instead of MDX code](#)
- [macro variables for conditional processing](#)
- [ability to open SAS Web Report Studio reports](#)
- [new and enhanced SAS tasks](#)

Contact Information

Larry Hoyle

Institute for Policy & Social Research,

University of Kansas

LarryHoyle@ku.edu

<http://www.ipsr.ku.edu/>