SAS® Programmers Guide for Version 9

David Shannon
Amadeus Software Limited
Topics Covered:

- Changes to format lengths
- New formats and informats
- New functions
- New SAS system options
- New data set options
- Object Dot Syntax
  - Hash Tables
- ARM Macros
- Extensible Markup Language (XML)
- The Output Delivery System
### New Formats, Functions & Macros

<table>
<thead>
<tr>
<th>FORMAT</th>
<th>V6 &amp; V8</th>
<th>V9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric Format</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>Numeric InFormat</td>
<td>7</td>
<td>31</td>
</tr>
<tr>
<td>Character Format</td>
<td>7</td>
<td>31</td>
</tr>
<tr>
<td>Character InFormat</td>
<td>6</td>
<td>30</td>
</tr>
</tbody>
</table>
New Formats, Functions & Macros

- **VALIDFMTNAME** = System Option

  - The VALIDFMTNAME controls which format names may be used in the current session

  ```
  options validfmtname= v8 | v9
  ```
New Formats, Functions & Macros

• **Any Date Informats**
  – Convert a string into the following formats:

  • ANYDTDTEw. SAS Date Value
  • ANYDTDTMw. SAS Date Time Value
  • ANYDTTMEw. SAS Time Value

  – It must be remembered these informats make assumptions on a record by record basis

See ANYDATE.SAS
New Formats, Functions & Macros

- **DATESTYLE option**
  - Sets a default assumption when ambiguous values in dates exist

  ```
  options datestyle = DMY | MDY | YMD
  ```

See ANYDATE.SAS
New Formats, Functions & Macros

- **PERL Regular Expressions & Pattern Matching**

  - Several new call routines and functions to perform pattern matching routines on data

  - A key function is the PRXPARSE function designed to specify pattern matching

  - See [www.perldoc.com](http://www.perldoc.com) for the complete list of expression values

See PRX.SAS
New Formats, Functions & Macros

Key statements

- Replacing the letter ‘z’ with letter ‘s’.

```sas
data _null_
    if _N_ = 1 then do;
        retain re;
        re = prxparse('s/z/s/');
    end;

    input;
    call prxchange(re, -1, _infile_);

    put _infile_;

datalines;
parenthesized
optimized
randomized
;
run;
```
New Formats, Functions & Macros

\[
\text{retain re;}
\]
\[
re = \text{prxparse('s/z/s/');}
\]
\[
\text{end,}
\]

- Specifies characters will be substituted
- Character to search for
- Character to use as replacement
New Formats, Functions & Macros

Column containing parse instructions

“-1” instructs the function to replace all occurrences

text to parse
New Formats, Functions & Macros

• **COUNT | COUNTC**
  - Returns the number of characters that do/do not exist in a character string

• **COMPARE**
  - Finds the first character where two strings do not match
  - Particularly useful for validating dual data entry systems
CALL SYMPUTX

- Allows conversion of numeric variable to a macro variable in a single call function
- Avoids numeric to character conversion issues
- Left justifies, trims and converts numeric to character for macro variables…
New Formats, Functions & Macros

```
data work.mvar;
  length numeric_var 8;
  numeric_var=1;
  * Note the need for 'put' and 'compress' statements on numeric variables;
  call syput('MyMacroVar',compress(put(numeric_var,??best.)));
run;
```

```
data work.mvar;
  length numeric_var 8;
  numeric_var=1;
  * In version 9 simply use call syputx;
  call syputx('MyMacroVar',numeric_var);
run;
```

- Numeric to Character Conversion
- Automatic Conversion
New Formats, Functions & Macros

• ANY and NOT Character Functions
  – Alphanumeric
  – Alphabetic
  – Control character
  – Digit
  – Character in a SAS variable name
  – Graphical character
  – Printable character
  – Punctuation mark
  – Space
  – Upper or lower character
  – Digit in a hex character
New Formats, Functions & Macros

• **CPUCOUNT**
  – Limits the number of processors used in threading procedures
  – Default is 0 which denotes all CPU’s

• **THREADS | NOTHREADS**
  – WinNT option to turn off multi-threading
  – Default is THREADS
New Formats, Functions & Macros

- **DTRESET**
  - Updates the date and time in the log and listing file

SAS V8 Keeps Session Start Time
New Formats, Functions & Macros

• Applying the DTRESET system option

```sas
options date dtreset;
ods rtf file='c:\sasdateupdate.rtf';
  proc print data=sashelp.vmacro;
    run;
  ods rtf close;
```
 ROLE Option for Proc SQL

- When a data set is being used in a star schema join, the table can be labelled FACT or DIMENSION.
- This can speed up processing during SQL joins
New Formats, Functions & Macros

• ROLE Option for Proc SQL

```sql
proc sql;
select a.manager, a.title, a.dept, b.employee
from saved.managers(role=fact) a,
    saved.employee(role=dim) b
where a.dept=b.dept;
quid;
```

Use of FACT and DIM options in a star schema design
New Formats, Functions & Macros

- **Object Dot Syntax**
  - Data step syntax has been extended, introducing Dot Notation (similarly to V8 SCL and several other programming languages)

  - The DECLARE statement allows the creation of an object, methods may then be called on that object:

    ```
    object.method();
    ```
New Formats, Functions & Macros

• Hash Tables

– Use of dot notation syntax in V9
– Hash tables are thoroughly documented in the academic world of computer sciences
– Extensively used in programming languages such as C++ and the newest Microsoft.NET languages C# and VB.
– Hash tables are lookup tables where all information is stored only in memory (rather than on disk)
New Formats, Functions & Macros

• Hash Tables Create Similar Results to:
  – Proc Format
  – Macro variables
  – Arrays
  – SQL joins
  – Indexing
  – Data step merges
New Formats, Functions & Macros

• Hash Table Syntax

```plaintext
declare associativearray aa(dataset: "saved.managers");
aa.defineKey('DEPT');
aa.defineData('MANAGER', 'TITLE');
aa.defineDone();
```

Declare type of object with name “aa”

DefineKey method to set the lookup key

DefineData method tells SAS we’re done

DefineKey method to set the lookup key
New Formats, Functions & Macros

- Hash Table Syntax

```sas
data work.hash;
length manager $20 title $50;
if _n_ = 1 then do;
    declare associativearray aa(dataset: "saved.managers");
    aa.defineKey('DEPT');
    aa.defineData('MANAGER','TITLE');
    aa.defineDone();
end;
set saved.employee;
if aa.find() = 0;
run;
proc print data=work.hash;
run;
```

Performs the lookup on the key DEPT
New Formats, Functions & Macros

• Hash Table Performance

  – Test performed on V9 EA for Windows
  – Pentium 4, 1.5GHz, 512MB RAM with Windows XP
  – Master data set has 1 million key/data pairs
  – Transaction data set has 1 million search keys
  – Half the search keys exist in the master data set
New Formats, Functions & Macros

- Hash Table Performance

<table>
<thead>
<tr>
<th>Format and Put() statements:</th>
<th>22.84 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET with KEY= statements:</td>
<td>47.85 seconds</td>
</tr>
<tr>
<td>MERGE with BY statements:</td>
<td>9.18 seconds</td>
</tr>
<tr>
<td>Hash Table:</td>
<td>3.43 seconds</td>
</tr>
</tbody>
</table>
New Formats, Functions & Macros

- Application Response Measurement (ARM) Macros
  - Provide industry standard output for ARM applications to read
  - Enhanced macros for version 9
  - Allow application developers to perform benchmarking of SAS programs
New Formats, Functions & Macros

- ARM Macros
  - Found in Base SAS Autocall macro library

- ARMEND
- ARMEND2
- ARMJOIN
- ARMINIT
- ARMINIT2
- ARMSTR
- ARMSTR2
- ARMGTID
- ARMGTID2
- ARMPROC
- ARMSTO
- ARMSTP2
- ARMUPD
- ARMUPD2
- ARMINI2
- ARMSTRT
Procedures, XML & ODS

• **Extensible Markup Language (XML)**
  – XML is simply text
  – Highly flexible and customizable structure for storing and transferring data
  – Makes moving data between systems particularly flexible
  – Main disadvantage is that it can contain a lot of matter, relative to the size of data
• Extensible Markup Language (XML)
  – Maturing use in industry:
    • Internet
    • .NET Microsoft Technologies
    • CDISC
  – SAS V8 does not handle XML at all well
  – SAS V9 introduces XMLMAP to interpret non-standard XML data for reading into SAS
Extensible Markup Language (XML)

To read XML:
1. Understand the form of our XML
2. Create a MAP file to interpret the layout of our XML
3. A reference to our map file is used on the libname statement
Procedures, XML & ODS

• XML

  – Example:
  Read into a SAS data set some NHL hockey team data:

```xml
<?xml version="1.0" encoding="iso-8859-1" ?>
<NHL>
  <CONFERENCE>
    Eastern
    <DIVISION>
      Southeast
      <TEAM name="Thrashers" abbrev="ATL" />
      <TEAM name="Hurricanes" abbrev="CAR" />
      <TEAM name="Panthers" abbrev="FLA" />
      <TEAM name="Lightning" abbrev="TB" />
      <TEAM name="Capitals" abbrev="WSH" />
    </DIVISION>
  </CONFERENCE>
  <CONFERENCE>
    Western
    <DIVISION>
      Pacific
      <TEAM name="Stars" abbrev="DAL" />
      <TEAM name="Kings" abbrev="LA" />
      <TEAM name="Ducks" abbrev="ANA" />
      <TEAM name="Coyotes" abbrev="PHX" />
      <TEAM name="Sharks" abbrev="SJ" />
    </DIVISION>
  </CONFERENCE>
</NHL>
```
<?xml version="1.0" ?>

<SXLEMAP version="1.1">
<TABLE name="teams">
   <TABLE_XPATH>
      /NHL/CONFERENCE/DIVISION/TEAM
   </TABLE_XPATH>

   <COLUMN name="name">
      <XPATH>
         /NHL/CONFERENCE/DIVISION/TEAM/@name
      </XPATH>
      <TYPE>Character</TYPE>
      <DATATYPE>STRING</DATATYPE>
      <LENGTH>30</LENGTH>
   </COLUMN>

   <COLUMN name="abbrev">
      <XPATH>
         /NHL/CONFERENCE/DIVISION/TEAM/@abbrev
      </XPATH>
      <TYPE>Character</TYPE>
      <DATATYPE>STRING</DATATYPE>
      <LENGTH>3</LENGTH>
   </COLUMN>

   <COLUMN name="CONFERENCE" retain="YES">
      <XPATH>
         /NHL/CONFERENCE
      </XPATH>
      <TYPE>Character</TYPE>
      <DATATYPE>STRING</DATATYPE>
      <LENGTH>10</LENGTH>
   </COLUMN>

   <COLUMN name="DIVISION" retain="YES">
      <XPATH>
         /NHL/CONFERENCE
      </XPATH>
      <TYPE>Character</TYPE>
      <DATATYPE>STRING</DATATYPE>
      <LENGTH>10</LENGTH>
   </COLUMN>

</TABLE>
</SXLEMAP>
Procedures, XML & ODS

- XML

```sas
filename map "nhl.map";
libname nhl xml "nhl.xml" xmlmap=map;
proc print data=nhl.teams noobs;
run;
```
Procedures, XML & ODS

- **XML**
  - To aid us in creating our XMLMAP files SAS is producing an application called XML Atlas
  - XML Atlas is an environment which allows the user to import XML files, then through drag-and-drop functionality we can build the XMLMAP interactively.
Procedures, XML & ODS

• The Output Delivery System

– Greater support for markup languages, you can now create your own!

– New ODS DOCUMENT for storing, enhancing and replaying ODS generated output

– Enhancements and refinements to existing ODS destinations.
Procedures, XML & ODS

- **ODS CHTML**
  - Creates Compact HTML possible (fewest tags) without using styles

```sas
chtml.sas
ods chtml file="vmacro.html"
proc print data=sashelp.vmacro;
run;
ods chtml close;
```
Procedures, XML & ODS

- **ODS CSV**
  - Creates a comma delimited CSV file of table information
  - Typically imported into Microsoft Excel

- **ODS CSVALL**
  - Creates a comma delimited CSV file of table information, preserving titles, notes and by lines.
Procedures, XML & ODS

• **ODS DOCBOOK**
  - Creates XML files supporting the DocBook DTD format from Oasis

• **ODS IMODE**
  - Produces HTML in column form that is separated by lines and lists all columns in rows.
Procedures, XML & ODS

- **ODS LaTeX**
  - Produces LaTeX output for high quality typesetting systems

- **ODS PCL**
  - Produces information for HP LaserJet emulation

- **ODS TROFF**
  - Produces Troff markup language for high quality laser printing and typesetting systems
Procedures, XML & ODS

• **ODS DOCUMENT**
  - Sends any ODS output to a document store (SAS Item Store)

See DOCUMENT.SAS
• ODS DOCUMENT

- ODS output may then be replayed without regenerating the output
- ODS output may also have its styles altered
- PROC DOCUMENT is used to replay the output to an ODS destination
- See also View pull-down menu from the results window
Procedures, XML & ODS

• **ODS HTML**
  – In SAS V9.1 the HTML standard produced by ODS HTML will be W3C’s HTML4.
  – The current standard will be available via ODS MARKUP

• **ODS HTMLCSS**
  – Generates a CSS file from a SAS style sheet
  – HTMLCSS will also use an existing CSS if one is specified
ODS HTML for Microsoft Excel (SAS V8 tip)

- As Microsoft 97 and newer, recognises HTML, we can create a HTML file with a XLS extension.
- Windows then assumes this file to be an Excel workbook and launches Excel to read the file:

```sas
ods html file="class.xls" style=default;
proc print data=sashelp.class;
run;
ods html close;
```
The SAS System

<table>
<thead>
<tr>
<th>Obs</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69</td>
<td>112.5</td>
</tr>
<tr>
<td>2</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84</td>
</tr>
<tr>
<td>3</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98</td>
</tr>
<tr>
<td>4</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>5</td>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
<tr>
<td>6</td>
<td>James</td>
<td>M</td>
<td>12</td>
<td>57.3</td>
<td>83</td>
</tr>
<tr>
<td>7</td>
<td>Jane</td>
<td>F</td>
<td>12</td>
<td>59.8</td>
<td>84.5</td>
</tr>
<tr>
<td>8</td>
<td>Janet</td>
<td>F</td>
<td>15</td>
<td>62.5</td>
<td>112.5</td>
</tr>
<tr>
<td>9</td>
<td>Jeffrey</td>
<td>M</td>
<td>13</td>
<td>62.5</td>
<td>84</td>
</tr>
<tr>
<td>10</td>
<td>John</td>
<td>M</td>
<td>12</td>
<td>59</td>
<td>99.5</td>
</tr>
<tr>
<td>11</td>
<td>Joyce</td>
<td>F</td>
<td>11</td>
<td>51.3</td>
<td>50.5</td>
</tr>
<tr>
<td>12</td>
<td>Judy</td>
<td>F</td>
<td>14</td>
<td>64.3</td>
<td>90</td>
</tr>
<tr>
<td>13</td>
<td>Louise</td>
<td>F</td>
<td>12</td>
<td>56.3</td>
<td>77</td>
</tr>
<tr>
<td>14</td>
<td>Mary</td>
<td>F</td>
<td>15</td>
<td>66.5</td>
<td>112</td>
</tr>
<tr>
<td>15</td>
<td>Philip</td>
<td>M</td>
<td>16</td>
<td>72</td>
<td>150</td>
</tr>
<tr>
<td>16</td>
<td>Robert</td>
<td>M</td>
<td>12</td>
<td>64.8</td>
<td>128</td>
</tr>
<tr>
<td>17</td>
<td>Ronald</td>
<td>M</td>
<td>15</td>
<td>67</td>
<td>133</td>
</tr>
<tr>
<td>18</td>
<td>Thomas</td>
<td>M</td>
<td>11</td>
<td>57.5</td>
<td>85</td>
</tr>
<tr>
<td>19</td>
<td>William</td>
<td>M</td>
<td>15</td>
<td>66.5</td>
<td>112</td>
</tr>
</tbody>
</table>
Procedures, XML & ODS

- **ODS WML**
  - Creates Wireless Markup Language WAP based environments

- **ODS WMLOLIST**
  - As WML but also creates a table of contents option list
Procedures, XML & ODS

• ODS MARKUP

  – Allows custom tag set creation
  
  – Any customised tag sets can be created, registered within SAS and used
  
  – There are several built in Tag Sets
proc template;
  define tagset Tagsets.Wml;
    notes "This is the WML definition";
  define event colspecs;
    put "<p>" NL;
    put "<table>";
    putq " columns=" COLCOUNT;"
    put "</>" NL;
  end;
  define event table;
    finish:
      put "</table>" NL;
      put "</p>" NL;
  end;
  define event row;
    start:
      put "<tr>";
      put "</tr>" NL;
    finish:
      put "</tr>" NL;
  end;
  define event header;
    start:
      put "<td><strong>";
      put VALUE;
      finish:
      put "</strong></td>";
    end;
  define event data;
    start:
      put "<td>";
      put VALUE;
    finish:
      put "</td>";
    end;
  define event colspanfill;
    put "<td/>>";
  end;
  define event rowspanfill;
    put "<td/>>";
  end;
  define event breakline;
    put "<br/>";
  end;
  define event splitline;

Procedures, XML & ODS

• New ODS Options

  – COLUMNS
    • Creates multicolumn output in RTF and PDF destinations
• New ODS Options

  – MARGIN and INDENT
    • Controls the margins and indentation of cells in RTF and PDF destinations

```
define comment4 / computed
  style={leftmargin=12pt indent=-12pt}
  'LeftMargin reverse(negative) indent';
```
Procedures, XML & ODS

- **New ODS Options**
  - **TEXT option**
    - Experimental in V8 and produced several undesirable results, but is now production in V9:

```sas
title "Text Demonstration";
ods rtf file='c:\v9\text.rtf'
  startpage=no
  text='Text before the report (not a title).';
proc print data=sashelp.vmacro;
run;
ods rtf text='Text after the report (not a footnote).';
```
Procedures, XML & ODS

• New ODS Options
  – Page X of Y Support
  • Ways of embedding raw RTF in V8, wrapped up into a single option in V9

```sas
ods escapechar = '\';
title1 'Page X of Y example.' j=r 'Page \{pageof}';
title2 This works in RTF only.;
ods rtf file='c:\pagexofy.rtf';
proc print data=sashelp.vmacro;
run;
ods rtf close;
```
Procedures, XML & ODS

- **New ODS Options**
  - Decimal Alignment
    - Automatically aligns decimals in columns for the RTF destination

```sas
title Decimal Aligned Listing;
proc print data=work_decimal;
  var charvar numvar / style(COLUMN)={just=d}
  ;
run;
ods rtf close;
```
Procedures, XML & ODS

• **New ODS Options**
  – **Orientation**
  • Allows RTF documents to change page orientation mid way through a document

```sas
options orientation = landscape;
ods rtf file = 'c:\v9\portrait.Rtf';
proc print data = sashelp.vmacro;
  run;
options orientation = portrait;
* Notify ODS RTF that the orientation is now portrait;
ods rtf;
proc print data = sashelp.vmacro;
  run;
ods rtf close;
```
Procedures, XML & ODS

• ODS with SAS/GRAPH
  – The ANNOTATE option is now supported in ODS with most SAS Graph procedures.
  – Colours may now be 64bytes long
  – Server side rendering is available with the ACTXIMG and JAVAIMG device drivers. These support ActiveX and Java, respectively.
**SAS/Connect**

- **Cross Environment Data Access (CEDA)**
  - Still supported in V9 either by PROC UPLOAD / DOWNLOAD or by creating a transport file
  - CEDA automatically converts between 32bit and 64bit data on different operating systems
  - In V9, the SAS log lists messages from CEDA.
SAS/Connect

• LIBREF Inheritance
  – Client session defined librefs are now inherited and do not have to be reassigned in multiple server session environments

• %SYSLPUT CONNECTREMOTE= Option
  – Macro variables may be created on remote session. The CONNECTREMOTE= option specifies the session name
Questions on Programming SAS System 9?

David Shannon
Senior Consultant
Amadeus Software Limited
Email: david.shannon@amadeus.co.uk
Web: www.amadeus.co.uk