

Version 9 Changes and Enhancements



Global Business Intelligence Consulting and Training

Destiny Corporation • 100 Great Meadow Rd Suite 601
Wethersfield, CT 06109-2379
Phone: (860) 721-1684 • 1-800-7TRAINING
Fax: (860) 721-9784
Email: info@destinycorp.com Web Site:
www.destinycorp.com

Prepared by Dana Rafiee

Copyright ©2002 Destiny Corporation. All Rights Reserved.

Introduction

This document is an excerpt from Destiny Corporation's Version 9 Changes and Enhancement course materials. It is designed to give a brief overview of what Version 9 has to offer.

Version 9 of the SAS System will roll out in several releases. The release date of Version 9.0 is mid 2002 and Version 9.1 will be the beginning of 2003. Some of the enhancements and features discussed in this course will be first available in either 9.0 or 9.1.

The primary goal of Version 9 is to provide support for a new level of computing that supports faster execution of applications, centralized access of data and support of the latest computing technology. These materials are designed as an overview of what is available and new and complement the online documentation that ships with the software.

The new SAS Open Metadata Architecture is designed to allow all registered data in an organization to be centrally managed and accessed. This feature will ship with Version 9.1.

The SAS Management Console will offer one point of control for all SAS servers and applications in the organization.

The new multi-threaded architecture is automatically turned on to support most procedures that sort and summarize data.

The Output Delivery System has been enhanced to support many new styles of markup, along with custom markup tag sets.

SAS also supports the industry standard Application Response Measurement (ARM) protocol for monitoring SAS processes.

SAS V9 is designed to be upwardly compatible with code and data.

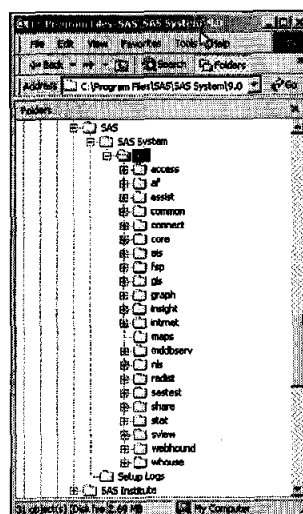
Cross Environment Data Access (CEDA) is still supported on all operating systems.

SAS Version 9 is designed to better support ADA 508 handicapped standards.

This course is designed to discuss as many issues as possible, from the installation of SAS to the finer aspects of operating under the new release.

Installation of SAS

SAS is now installed in a new directory location. The following example shows the tree structure.



The new directory for Version 9 is SAS. SAS V9 is designed to live alongside SAS V8. Notice the SAS V9 directory and the SAS Institute V8 directory.

Note: SAS V9 and SAS V8 use some of the same shared components like the Enhanced Editor and ActiveX controls. When uninstalling one of the versions, do not uninstall the shared components.

When installing SAS V9, the footprint on disk may be large. You may consider deleting the SAS Video files and Maps that are not needed at your installation.

Configuration File Changes

The configuration file uses pointers to different locations than found in previous versions. It is important to be aware

[illegible]

SETINIT Changes

When installing SAS, there is now a selection for selecting all products licensed by the click of a button.

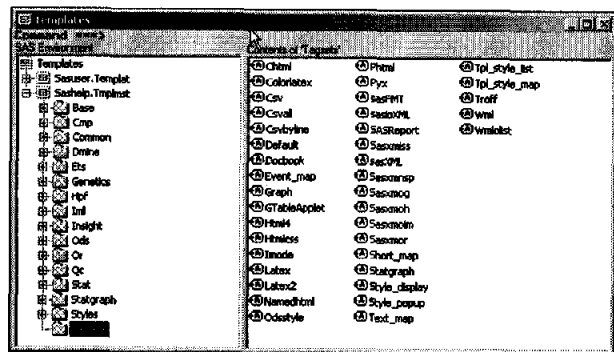
Display Manager Changes

List View Copy Command – DMCOPYLSV

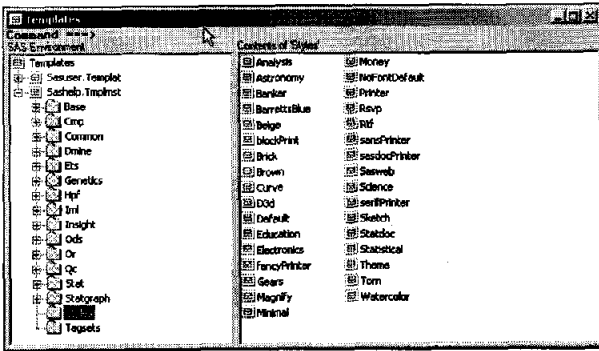
Name	Ending	Type	Host Path Name	Root
Sashelp	V9	Library	C:\Program Files\SAS\SAS System\3 Objects\SASCFG\C:\Program Files\SAS\SAS System\3 Objects	
Maps	V9	Library	C:\Program Files\SAS\SAS System\3 Objects	
Sasuser	V9	Library	C:\Documents and Settings\drafae\DESTINY\COMPANY\DATA	
Work	V9	Library	C:\DOCUMENT-1\DRAFIE-1\DES\LOCALS-1\Temp\SAS Temp...	

"Name", "Engine", "Type", "Host Path Name", "Modified",
"Sashelp", "V9", "Library", "('C:\Program Files\SAS\SAS
System\9.0\insent\SASCFG' 'C:\Program Files\SAS\SAS
System\9.0\core\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\af\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\assist\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\connect\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\eis\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\gis\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\graph\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\insight\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\intrnet\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\mdbdbserv\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\sviwe\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\stat\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\whouse\sashelp' 'C:\Program Files\SAS\SAS
System\9.0\webhound\sashelp')", "",
"Maps", "V9", "Library", "C:\Program Files\SAS\SAS
System\9.0\maps", "",
"Sasuser", "V9", "Library", "C:\Documents and
Settings\drafee.DESTINYCORP\My Documents\My SAS
Files\V9", "",
"Work", "V9", "Library",
"C:\DOCUME~1\DRAFIE~1\DES\LOCALS~1\Temp\SAS
Temporary Files\TD1488", ""

Select Tagsets from the Templates selection in the Results window to see all of the tagsets written and included with SAS software.

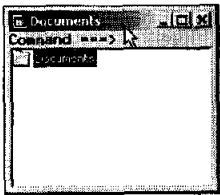


279



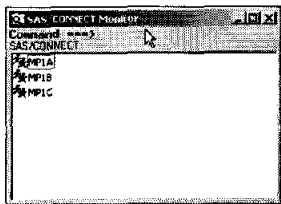
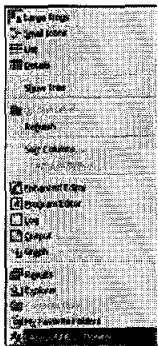
ODS Document Viewer

The new ODS document viewer is now available. It can be used to see stored documents created with the ODS DOCUMENT statement. See the ODS section of this document for further details.

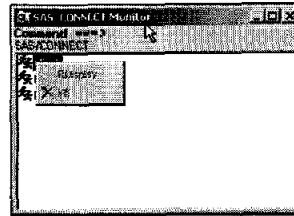


SAS/Connect Monitor

This option is available from the Tools pull down menu. It allows for monitoring of running SAS/Connect sessions.



Right clicking on each session allows for killing the session or viewing output.



Return of the V6 Windows

SAS brought back some of the old Version 6 windows in Display Manager per the request of users.

The commands to access them are:

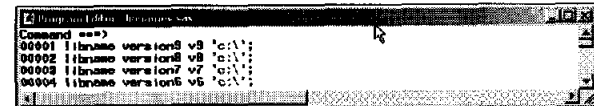
V6CAT
V6LIB
V6DIR
V6VAR

Data Migration

In Version 9, SAS supports formats and informats that are longer than 8 bytes. The only difference in the data set structures is this capability. Version 8 data sets are upward compatible to Version 9 by default. Version 9 data sets are backward compatible if formats and informats conform to Version 8 naming conventions.

Engines

Libname statements offer support for the new V9 engine as well as the V8, V7 and V6 engines.



The file extensions are still the same as in Version 8.

File Extension	SAS Member Type	Description
.log	None	Log
.lst	None	Output
.sas	None	SAS Program
.sas7bacs	Access	Access descriptor
.sas7baud	Audit	Audit file
.sas7bcatalog	Catalog	SAS catalog
.sas7bdat	Table	Data set
.sas7bmdb	MDDb	Multi-dimensional database

.sas7bndx	None	Data set index. Indexes are stored as separate files but are treated by the SAS System as integral parts of the SAS data file.
.sas7bods		Output Delivery System file
.sas7bpqm	Program	Stored DATA step program
.sas7bvew	View	Data set view

Catalog Migration

Note: Version 6 catalogs or Version 8 catalogs on 32 bit systems that are being upgraded to 64 bit systems must be updated from previous versions of SAS to Version 9 with the standing CPORT/CIMPORT procedures.

```

Program Editor - cport.sas
Command ==>
00001 filename tranfile 'c:\tranfile';
00002 libname old v8 'c:\data\sas\data\';
00003 libname new v9 'c:\v9';
00004 proc cport - library=old files=tranfile;
00005 select class$B;
00006 run;
00007 proc cimport library=new files=tranfile;
00008 select class$B;
00009 run;
00010 run;

```

New Formats

The new length for numeric format names is 32 and for character names is 31. Format names can be associated with a data set. The only difference between SAS V9 and previous versions of data sets is the existence of a format name longer than the traditional 8 bytes.

Version 9 data sets that use 8 byte or shorter format names can be read by Version 8.

\$BIDw.

This format is designed to convert a string to be logically or visually ordered. This works with Hebrew and Latin characters.

New Informats

The new length for numeric informat names is 31 and for character names is 30. Informat names can be associated with a data set. The only difference between SAS V9 and previous versions of data sets is the existence of a format name longer than the traditional 8 bytes.

Version 9 data sets that use 8 byte or shorter format names can be read by Version 8.

Any Date Informat

Three new informats are now available to convert various date, time and datetime forms of data into a SAS date or SAS time. They are:

ANYDTDTew. To convert to a SAS date value

ANYDTTMEw. To convert to a SAS time value
ANYDTDTMw. To convert to a SAS datetime value

These new informats were created to make reading these types of values simpler. It is important to realize that these informats make assumptions on a record by record basis. Ambiguous values can be interpreted in an incorrect fashion.

```

Program Editor - anydate.sas
Command ==>
00001 data work.dates;
00002 infile cards;
00003 input @01 string $ 20.;
00004 extracteddate = input(string,anydate32.);
00005 extractedtime = input(string,anytime32.);
00006 extracteddatetime = input(string,anydatetime32.);
00007 datelines;
00008 2002-Apr-15
00009 Apr 11 15, 2002
00010 Apr 11 15 2002
00011 200202
00012 15APR2002
00013 2002110
00014 2002/04/15
00015 2002/04/15:11:22:33
00016 15APR2002:11:22:33
00017 12:34:56
00018 APR2002
00019 15/04/2002
00020 04/15/2002
00021 run;
00022 proc print data=work.dates;
00023 run;

```

Special Interpretations

Timezones with + or - GMT times will be ignored.
The YEARCUTOFF= option interprets two digit years

DATESTYLE Option

The DATESTYLE option can be used when ambiguous values in dates exist. This option sets a default assumption for the date to be either DMY, MDY, or YMD.

```

Program Editor - anydate.sas
Command ==>
00001 options datestyle=MDY;
00002 data work.dates;
00003 infile cards;
00004 input @01 string $ 20.;
00005 extracteddate = input(string,anydate32.);
00006 extractedtime = input(string,anytime32.);
00007 extracteddatetime = input(string,anydatetime32.);
00008 datelines;
00009 2002-Apr-15
00010 Apr 11 15, 2002
00011 Apr 11 15 2002
00012 200202
00013 15APR2002
00014 2002110
00015 2002/04/15
00016 2002/04/15:11:22:33
00017 15APR2002:11:22:33
00018 12:34:56
00019 APR2002
00020 15/04/2002
00021 04/15/2002
00022 run;
00023 proc print data=work.dates;
00024 run;

```

New Functions

There are several, new functions and call routines available in SAS. Most of these are designed for very specific manipulations of data.

Perl Regular Expressions and Pattern Matching

There are several functions available to perform pattern matching routines on data. The following example shows a typical use for one of these functions. The PRXPARSE function is designed to specify pattern matching. This example sets YES/NO flags as to how phone numbers match a valid pattern.

ANYUPPER	Finds the first occurrence of an upper case character
ANYXDIGIT	Finds the first occurrence of a digit in a hex character
CALL SORTQ	Sorts values of variables in the Data Step
CAT, CATS, CATT, CATX	Concatenate passed parameters without the need for using TRIM, LEFT and PUT functions.
COMPARE	Finds the first occurrence of where two strings differ
COMPGED	Uses generalized edit distance to compare two strings
COMPLEV	Uses Levenshtein edit distance to compare two strings
COUNT	Returns the number of occurrences of a character in a string
COUNTC	Returns the number of occurrences of a character in a string that exist or do not exist
FIND	Like INDEX function searches, but not case or direction sensitive
FINDC	Like INDEXC function searches, but not case or direction sensitive
LENGTHC	Finds the length of a character value including trailing blanks
LENGTHM	Finds the length used by a character value in memory
LENGTHN	Finds the length of a character value excluding trailing blanks
NINVALID	Determines if the string is a valid SAS name
NLITERAL	Converts a string to a literal if not a valid SAS name
NOTALNUM	Finds the first occurrence of a non alphabetic character
NOTCNTRL	Finds the first occurrence of a non control character
NOTDIGIT	Finds the first occurrence of a non digit
NOTFIRST	Finds the first position of a character value that could be used as the beginning of a SAS name
NOTGRAPH	Finds the first occurrence of a non graphical character
NOTLOWER	Finds the first occurrence of a non lower case character
NOTNAME	Finds the first occurrence of a non valid character for a SAS name
NOTPRINT	Finds the first occurrence of a non printable character
NOTPUNCT	Finds the first occurrence of a non punctuation character
NOTSPACE	Finds the first occurrence of a non white space character
NOTUPPER	Finds the first occurrence of a non upper case character
NOTXDIGIT	Finds the first occurrence of a non digit in a hex value

NVALID	Determines if a string can be used as a SAS name
SCANQ	Finds a word in a string while ignoring delimiters inside quotes
SUBSTRN	Allows for doing a substring where the length may be zero

CALL SORTQ Example

CALL SORTQ is a quick way to sort variable values inside the Data Step. It is not designed to replace PROC SORT. It is a simple way of ordering values of the same structure. For example, if it is used with character variables, they must be the same length.

Consider the following example. Four variables are loaded into a Data Step. We want them ordered smallest to largest to process on.

```

1  * Program listing: callsort.sas
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *
67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *
181 *
182 *
183 *
184 *
185 *
186 *
187 *
188 *
189 *
190 *
191 *
192 *
193 *
194 *
195 *
196 *
197 *
198 *
199 *
200 *
201 *
202 *
203 *
204 *
205 *
206 *
207 *
208 *
209 *
210 *
211 *
212 *
213 *
214 *
215 *
216 *
217 *
218 *
219 *
220 *
221 *
222 *
223 *
224 *
225 *
226 *
227 *
228 *
229 *
230 *
231 *
232 *
233 *
234 *
235 *
236 *
237 *
238 *
239 *
240 *
241 *
242 *
243 *
244 *
245 *
246 *
247 *
248 *
249 *
250 *
251 *
252 *
253 *
254 *
255 *
256 *
257 *
258 *
259 *
260 *
261 *
262 *
263 *
264 *
265 *
266 *
267 *
268 *
269 *
270 *
271 *
272 *
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *
327 *
328 *
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *
727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *
1001 *
1002 *
1003 *
1004 *
1005 *
1006 *
1007 *
1008 *
1009 *
1010 *
1011 *
1012 *
1013 *
1014 *
1015 *
1016 *
1017 *
1018 *
1019 *
1020 *
1021 *
1022 *
1023 *
1024 *
1025 *
1026 *
1027 *
1028 *
1029 *
1030 *
1031 *
1032 *
1033 *
1034 *
1035 *
1036 *
1037 *
1038 *
1039 *
1040 *
1041 *
1042 *
1043 *
1044 *
1045 *
1046 *
1047 *
1048 *
1049 *
1050 *
1051 *
1052 *
1053 *
1054 *
1055 *
1056 *
1057 *
1058 *
1059 *
1060 *
1061 *
1062 *
1063 *
1064 *
1065 *
1066 *
1067 *
1068 *
1069 *
1070 *
1071 *
1072 *
1073 *
1074 *
1075 *
1076 *
1077 *
1078 *
1079 *
1080 *
1081 *
1082 *
1083 *
1084 *
1085 *
1086 *
1087 *
1088 *
1089 *
1090 *
1091 *
1092 *
1093 *
1094 *
1095 *
1096 *
1097 *
1098 *
1099 *
1100 *
1101 *
1102 *
1103 *
1104 *
1105 *
1106 *
1107 *
1108 *
1109 *
1110 *
1111 *
1112 *
1113 *
1114 *
1115 *
1116 *
1117 *
1118 *
1119 *
1120 *
1121 *
1122 *
1123 *
1124 *
1125 *
1126 *
1127 *
1128 *
1129 *
1130 *
1131 *
1132 *
1133 *
1134 *
1135 *
1136 *
1137 *
1138 *
1139 *
1140 *
1141 *
1142 *
1143 *
1144 *
1145 *
1146 *
1147 *
1148 *
1149 *
1150 *
1151 *
1152 *
1153 *
1154 *
1155 *
1156 *
1157 *
1158 *
1159 *
1160 *
1161 *
1162 *
1163 *
1164 *
1165 *
1166 *
1167 *
1168 *
1169 *
1170 *
1171 *
1172 *
1173 *
1174 *
1175 *
1176 *
1177 *
1178 *
1179 *
1180 *
1181 *
1182 *
1183 *
1184 *
1185 *
1186 *
1187 *
1188 *
1189 *
1190 *
1191 *
1192 *
1193 *
1194 *
1195 *
1196 *
1197 *
1198 *
1199 *
1200 *
1201 *
1202 *
1203 *
1204 *
1205 *
1206 *
1207 *
1208 *
1209 *
1210 *
1211 *
1212 *
1213 *
1214 *
1215 *
1216 *
1217 *
1218 *
1219 *
1220 *
1221 *
1222 *
1223 *
1224 *
1225 *
1226 *
1227 *
1228 *
1229 *
1230 *
1231 *
1232 *
1233 *
1234 *
1235 *
1236 *
1237 *
1238 *
1239 *
1240 *
1241 *
1242 *
1243 *
1244 *
1245 *
1246 *
1247 *
1248 *
1249 *
1250 *
1251 *
1252 *
1253 *
1254 *
1255 *
1256 *
1257 *
1258 *
1259 *
1260 *
1261 *
1262 *
1263 *
1264 *
1265 *
1266 *
1267 *
1268 *
1269 *
1270 *
1271 *
1272 *
1273 *
1274 *
1275 *
1276 *
1277 *
1278 *
1279 *
1280 *
1281 *
1282 *
1283 *
1284 *
1285 *
1286 *
1287 *
1288 *
1289 *
1290 *
1291 *
1292 *
1293 *
1294 *
1295 *
1296 *
1297 *
1298 *
1299 *
1300 *
1301 *
1302 *
1303 *
1304 *
1305 *
1306 *
1307 *
1308 *
1309 *
1310 *
1311 *
1312 *
1313 *
1314 *
1315 *
1316 *
1317 *
1318 *
1319 *
1320 *
1321 *
1322 *
1323 *
1324 *
1325 *
1326 *
1327 *
1328 *
1329 *
1330 *
1331 *
1332 *
1333 *
1334 *
1335 *
1336 *
1337 *
1338 *
1339 *
1340 *
1341 *
1342 *
1343 *
1344 *
1345 *
1346 *
1347 *
1348 *
1349 *
1350 *
1351 *
1352 *
1353 *
1354 *
1355 *
1356 *
1357 *
1358 *
1359 *
1360 *
1361 *
1362 *
1363 *
1364 *
1365 *
1366 *
1367 *
1368 *
1369 *
1370 *
1371 *
1372 *
1373 *
1374 *
1375 *
1376 *
1377 *
1378 *
1379 *
1380 *
1381 *
1382 *
1383 *
1384 *
1385 *
1386 *
1387 *
1388 *
1389 *
1390 *
1391 *
1392 *
1393 *
1394 *
1395 *
1396 *
1397 *
1398 *
1399 *
1400 *
1401 *
1402 *
1403 *
1404 *
1405 *
1406 *
1407 *
1408 *
1409 *
1410 *
1411 *
1412 *
1413 *
1414 *
1415 *
1416 *
1417 *
1418 *
1419 *
1420 *
1421 *
1422 *
1423 *
1424 *
1425 *
1426 *
1427 *
1428 *
1429 *
1430 *
1431 *
1432 *
1433 *
1434 *
1435 *
1436 *
1437 *
1438 *
1439 *
1440 *
1441 *
1442 *
1443 *
1444 *
1445 *
1446 *
1447 *
1448 *
1449 *
1450 *
1451 *
1452 *
1453 *
1454 *
1455 *
1456 *
1457 *
1458 *
1459 *
1460 *
1461 *
1462 *
1463 *
1464 *
1465 *
1466 *
1467 *
1468 *
1469 *
1470 *
1471 *
1472 *
1473 *
1474 *
1475 *
1476 *
1477 *
1478 *
1479 *
1480 *
1481 *
1482 *
1483 *
1484 *
1485 *
1486 *
1487 *
1488 *
1489 *
1490 *
1491 *
1492 *
1493 *
1494 *
1495 *
1496 *
1497 *
1498 *
1499 *
1500 *
1501 *
1502 *
1503 *
1504 *
1505 *
1506 *
1507 *
1508 *
1509 *
1510 *
1511 *
1512 *
1513 *
1514 *
1515 *
1516 *
1517 *
1518 *
1519 *
1520 *
1521 *
1522 *
1523 *
1524 *
1525 *
1526 *
1527 *
1528 *
1529 *
1530 *
1531 *
1532 *
1533 *
1534 *
1535 *
1536 *
1537 *
1538 *
1539 *
1540 *
1541 *
1542 *
1543 *
1544 *
1545 *
1546 *
1547 *
1548 *
1549 *
1550 *
1551 *
1552 *
1553 *
1554 *
1555 *
1556 *
1557 *
1558 *
1559 *
1560 *
1561 *
1562 *
1563 *
1564 *
1565 *
1566 *
1567 *
1568 *
1569 *
1570 *
1571 *
1572 *
1573 *
1574 *
1575 *
1576 *
1577 *
1578 *
1579 *
1580 *
1581 *
1582 *
1583 *
1584 *
1585 *
1586 *
1587 *
1588 *
1589 *
1590 *
1591 *
1592 *
1593 *
1594 *
1595 *
1596 *
1597 *
1598 *
1599 *
1600 *
1601 *
1602 *
1603 *
1604 *
1605 *
1606 *
1607 *
1608 *
1609 *
1610 *
1611 *
1612 *
1613 *
1614 *
1615 *
1616 *
1617 *
1618 *
1619 *
1620 *
1621 *
1622 *
1623 *
1624 *
1625 *
1626 *
1627 *
1628 *
1629 *
1630 *
1631 *
1632 *
1633 *
1634 *
1635 *
1636 *
1637 *
1638 *
1639 *
1640 *
1641 *
1642 *
1643 *
1644 *
1645 *
1646 *
1647 *
1648 *
1649 *
1650 *
1651 *
1652 *
1653 *
1654 *
1655 *
1656 *
1657 *
1658 *
1659 *
1660 *
1661 *
1662 *
1663 *
1664 *
1665 *
1666 *
1667 *
1668 *
1669 *
1670 *
1671 *
1672 *
1673 *
1674 *
1675 *
1676 *
1677 *
1678 *
1679 *
1680 *
1681 *
1682 *
1683 *
1684 *
1685 *
1686 *
1687 *
1688 *
1689 *
1690 *
1691 *
1692 *
1693 *
1694 *
1695 *
1696 *
1697 *
1698 *
1699 *
1700 *
1701 *
1702 *
1703 *
1704 *
1705 *
1706 *
1707 *
1708 *
1709 *
1710 *
1711 *
1712 *
1713 *
1714 *
1715 *
1716 *
1717 *
1718 *
1719 *
1720 *
1721 *
1722 *
1723 *
1724 *
1725 *
1726 *
1727 *
1728 *
1729 *
1730 *
1731 *
1732 *
1733 *
1734 *
1735 *
1736 *
1737 *
1738 *
1739 *
1740 *
1741 *
1742 *
1743 *
1744 *
1745 *
1746 *
1747 *
1748 *
1749 *
1750 *
1751 *
1752 *
1753 *
1754 *
1755 *
1756 *
1757 *
1758 *
1759 *
1760 *
1761 *
1762 *
1763 *
1764 *
1765 *
1766 *
1767 *
1768 *
1769 *
1770 *
1771 *
1772 *
1773 *
1774 *
1775 *
1776 *
1777 *
1778 *
1779 *
1780 *
1781 *
1782 *
1783 *
1784 *
1785 *
1786 *
1787 *
1788 *
1789 *
1790 *
1791 *
1792 *
1793 *
1794 *
1795 *
1796 *
1797 *
1798 *
1799 *
1800 *
1801 *
1802 *
1803 *
1804 *
1805 *
1806 *
1807 *
1808 *
1809 *
1810 *
1811 *
1812 *
1813 *
1814 *
1815 *
1816 *
1817 *
1818 *
1819 *
1820 *
1821 *
1822 *
1823 *
1824 *
1825 *
1826 *
1827 *
1828 *
1829 *
1830 *
1831 *
1832 *
1833 *
1834 *
1835 *
1836 *
1837 *
1838 *
1839 *
1840 *
1841 *
1842 *
1843 *
1844 *
1845 *
1846 *
1847 *
1848 *
1849 *
1850 *
1851 *
1852 *
1853 *
1854 *
1855 *
1856 *
1857 *
1858 *
1859 *
1860 *
1861 *
1862 *
1863 *
1864 *
1865 *
1866 *
1867 *
1868 *
1869 *
1870 *
1871 *
1872 *
1873 *
1874 *
1875 *
1876 *
1877 *
1878 *
1879 *
1880 *
1881 *
1882 *
1883 *
1884 *
1885 *
1886 *
1887 *
1888 *
1889 *
1890 *
1891 *
1892 *
1893 *
1894 *
1895 *
1896 *
1897 *
1898 *
1899 *
1900 *
1901 *
1902 *
1903 *
1904 *
1905 *
1906 *
1907 *
1908 *
1909 *
1910 *
1911 *
1912 *
1913 *
1914 *
1915 *
1916 *
1917 *
1918 *
1919 *
1920 *
1921 *
1922 *
1923 *
1924 *
1925 *
1926 *
1927 *
1928 *
1929 *
1930 *
1931 *
1932 *
1933 *
1934 *
1935 *
1936 *
1937 *
1938 *
1939 *
1940 *
1941 *
1942 *
1943 *
1944 *
1945 *
1946 *
1947 *
1948 *
1949 *
1950 *
1951 *
1952 *
1953 *
1954 *
1955 *
1956 *
1957 *
1958 *
1959 *
1960 *
1961 *
1962 *
1963 *
1964 *
1965 *
1966 *
1967 *
1968 *
1969 *
1970 *
1971 *
1972 *
1973 *
1974 *
1975 *
1976 *
1977 *
1978 *
1979 *
1980 *
1981 *
1982 *
1983 *
1984 *
1985 *
1986 *
1987 *
1988 *
1989 *
1990 *
1991 *
1992 *
1993 *
1994 *
1995 *
1996 *
1997 *
1998 *
1999 *
2000 *
2001 *
2002 *
2003 *
2004 *
2005 *
2006 *
2007 *
2008 *
2009 *
2010 *
2011 *
2012 *
2013 *
2014 *
2015 *
2016 *
2017 *
2018 *
2019 *
2020 *
2021 *
2022 *
2023 *
2024 *
2025 *
2026 *
2027 *
2028 *
2029 *
2030 *
2031 *
2032 *
2033 *
2034 *
2035 *
2036 *
2037 *
2038 *
2039 *
2040 *
2041 *
2042 *
2043 *
2044
```

DOCTOC	Table of contents files for SAS online documentation
DRESET	Updates the date and time in the log and listing file
EMAILAUTHPROTOCOL	Sets up SMTP email authentication protocol
EMAILID=	Sets the default email ID
EMAILPW	Sets the default email password
ERRORBYABEND	Determines how SAS responds to a By group error
FONTSLC	Location of the SAS font file
HELPCNCMD	Tell SAS to use the English index for command line requests
HELPINDEX	Sets the location of the help index files
HELPTOC	Sets the location of the help table of contents files
LOGPARM	SAS log file control options
PAGEBREAKINITIAL	Used primarily for Unix to insert page breaks for the log and listing
QUOTELENMAX=	Writes a string maximum length warning to the log in quotes
SORTEQUALS=	Specifies to maintain order of identical values when sorting. SAS performs best when this is set to NO. There is a big degradation of performance when this is set to YES.
SORTSIZE=	The amount of memory allocated for sorting
TERMSTMT=	Opposite of INITCMD. This specifies the SAS statements to be executed at the end of a SAS session
TEXTURELOC=	ODS style location for textures and images
THREADS NOTHREADS	This option turns thread usage on or off. It is on by default.
TOOLSMENU NOTOOLSMENU	This turns the tools menu on and off.
VALIDDFMTNAME=	Specifies V9 or previous conventions for format name lengths
VIEWMENU NOVIEWMENU	This option turns the SAS menu on and off.

There are several new system options available in Version 9.1. These options primarily relate to the SAS Open Metadata Server. This server will be used as a one stop metadata location for all organization wide data registered.

Option	Definition
METAID=	Specifies the installation of SAS on the Open Metadata Server
METAPASS=	Specifies the default password for OMS

METAPORT=	Specifies the TCP/IP port for OMS
METAPROTOCOL=	Specifies the network protocol for OMS
METAREPOSITORY=	Specifies the metadata repository for OMS
METASERVER=	Specifies the OMS
METAUSER=	Specifies the default user for OMS

New Data Set Options

Option	Definition
ENCODING=	This is designed for multinational language support, typically used with SAS/SHARE environments where data is shared between different countries. One data set can be referenced in one character set for one country and a different character set for a different country. See the table below for the encoding values supported.
ROLE=	When the data set is being used in a star schema style join, this table can be labeled FACT or DIMENSION. This can speed up processing if the appropriate tables are labeled. SAS will use the role identification during SQL joins.
SORTSEQ=	Determines the collating sequencing during sorting.
SPILL=NO/YES	This is an option on Data Step Views that tells SAS to produce or not produce spill files when a view is opened for two pass mode. This reduces the amount of disk space required for a spill file. See SAS Documentation for further information on this efficiency and when to use it.

ROLE= Option for Optimized Joins

The following code is a simple example of how we can define particular tables as FACT or DIMENSION files.

```

Command ***
00001 proc sql;
00002   select a.manager, a.title, a.dept, b.employee
00003   from saved.manager(role=fact) a,
00004   saved.employee(role=dim) b
00005   where a.dept=b.dept;
00006 quit;

```

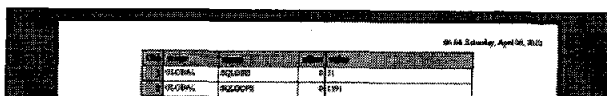
MANAGER	TITLE	DEPT	EMPLOYEE
B. DEAN	ADMINISTRATOR	301	AL FRANKLIN
B. DEAN	ADMINISTRATOR	301	CHARLIE HESTER
B. DEAN	ADMINISTRATOR	301	FAY TREWENT
J. JAMES	PRESIDENT	401	JOHN B. HEN
J. JAMES	PRESIDENT	401	JOHN DOE
J. JAMES	PRESIDENT	401	JOHN HENRY
J. JAMES	VICE PRESIDENT	301	JOHN DOE
B. DEAN	ADMINISTRATOR	301	LARRY BROWN
J. JAMES	PRESIDENT	401	PAUL JONES
J. JAMES	PRESIDENT	401	PAUL R. JONES
J. JAMES	VICE PRESIDENT	301	RICHARD BROWN
J. JAMES	PRESIDENT	401	STEVE RAY
B. DEAN	ADMINISTRATOR	301	STEVE WHITE

Resetting the SAS Date on Output

SAS introduces the DTRESET system option for resetting the date on SAS output.



```
Command window: sasdateupdate.sas
Command ***
00001 options date dtrset;
00002 ods rtf file='c:\sasdateupdate.rtf';
00003 proc print data=sashelp.vmacro;
00004 run;
00005 ods rtf close;
00006 quit;
```



DEPT	SQUARE	N
GLOBAL	SQUARE	2
GLOBAL	SQUARE	2

Putlog Statement

The new PUTLOG statement is designed to always write information to the SAS log, no matter where the FILEREF points to. This can be handy for debugging. It is similar to the PUT statement, but does not reference the FILEREF destination.

Object Dot Syntax

DECLARE Statement

The Data Step is being extended. SAS has introduced the concept of Object Dot Syntax. This is similar to the concept of Dot Notation as applied to Version 8 of SAS Component Language (SCL).

The DECLARE statement has been added to the data step, along with this syntax. This allows declaration of an object. The subsequent syntax allows for methods to be called on that object: Object.method()

Hash Table for Lookups

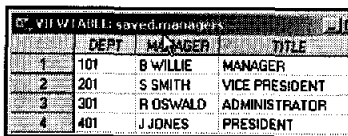
The first use of this new syntax has been introduced through Hash tables. Hash tables are a way of performing a table lookup by loading key variables and values into an array in memory and then matching those values to values being read from a data set.

The beauty of this feature is that the key information lives in memory and not on disk. This is another way of performing lookups that creates a similar result to using:

Proc format
Macro variables
Arrays
SQL Joins
Indexing
Data Step Merging

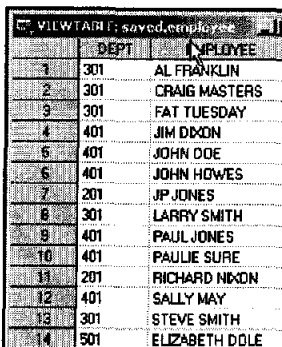
The hash table grows in memory based on the size of the data loaded. Consider the following example.

Managers Data Set



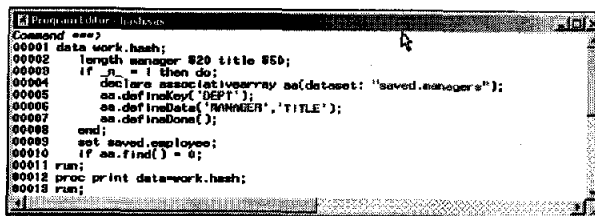
	DEPT	MANAGER	TITLE
1	101	B WILLIE	MANAGER
2	201	S SMITH	VICE PRESIDENT
3	301	R OSWALD	ADMINISTRATOR
4	401	J JONES	PRESIDENT

Employee Data Set



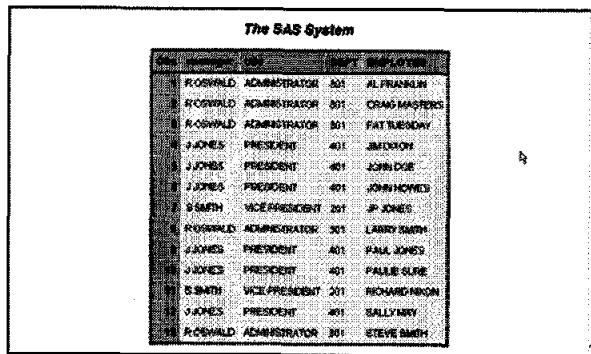
	DEPT	EMPLOYEE
1	301	AL FRANKLIN
2	301	CRAIG MASTERS
3	301	FAT TUESDAY
4	401	JIM DIXON
5	401	JOHN DOE
6	401	JOHN HOWES
7	201	JP JONES
8	301	LARRY SMITH
9	401	PAUL JONES
10	401	PAULIE SURE
11	201	RICHARD NIXON
12	401	SALLY MAY
13	301	STEVE SMITH
14	501	ELIZABETH DOLE

The following syntax will read both files.



```
Command window: hashlook
Command ***
00001 data work.hash;
00002 length manager $20 title $50;
00003 if _n_ = 1 then do;
00004   declare associativearray aa(dataset: "saved.managers");
00005   aa.definekey('DEPT');
00006   aa.defineref('MANAGER', 'TITLE');
00007   aa.defineDone();
00008 end;
00009 set saved.employees;
00010 if aa.find() = 0;
00011 run;
00012 proc print data=work.hash;
00013 run;
```

And yield the following result.



The SAS System			
R OSWALD	ADMINISTRATOR	301	AL FRANKLIN
R OSWALD	ADMINISTRATOR	301	CRAIG MASTERS
R OSWALD	ADMINISTRATOR	301	FAT TUESDAY
J JONES	PRESIDENT	401	JIM DIXON
J JONES	PRESIDENT	401	JOHN DOE
J JONES	PRESIDENT	401	JOHN HOWES
S SMITH	VICE PRESIDENT	201	JP JONES
R OSWALD	ADMINISTRATOR	301	LARRY SMITH
J JONES	PRESIDENT	401	PAUL JONES
J JONES	PRESIDENT	401	PAULIE SURE
S SMITH	VICE PRESIDENT	201	RICHARD NIXON
J JONES	PRESIDENT	401	SALLY MAY
R OSWALD	ADMINISTRATOR	301	STEVE SMITH

Multi-Threaded Architecture

One of the biggest enhancements with SAS software in Version 9 is its ability to support multi-threaded access to files for use in the Data Step and certain procedures. The concept is simple. Instead of using the traditional 'serial' approach to either sorting or summarizing data, SAS now breaks up the data into smaller chunks, performs the operation and then puts the result back together.

Sorting Analogy

Consider the following analogy for sorting. There are four decks of playing cards. The goal is to order them. One person can try to order all four decks together to produce an ordered result. Another way of doing this is to get four

people, each tasked with ordering one deck. The ordered four decks are then combined for a final result.

Summarizing Analogy

The goal is to get a total of 100 numbers. One person can sit down with a pencil and paper and total up the 100 numbers to produce a result. Another way of doing this is to get four people to each take 25 numbers and produce 4 totals. The 4 totals are then added up to produce the number.

The division of labor/tasks in these examples demonstrates why a multi-threaded architecture makes sense when possible.

Multi-threading is supported for:

PROC SORT
PROC SUMMARY
PROC MEANS
PROC REPORT
PROC TABULATE
PROC SQL
PROC REG
PROC GLM
PROC ROBUSTREG

By default, multi-threading is turned on in Version 9 for all of these procedures. Therefore, there is a new option available with each procedure (THREADS/NOTTHREADS) to optionally turn this feature off.

CPUCOUNT Option

It is important to realize that multi-threading works best in a multiple processor environment. For example, if SAS is running on a four processor server, it will attempt to utilize all of the CPUs available. For large SAS jobs, this may impact performance of other applications or programs running on that server.

The default value of CPUCOUNT is set to the maximum number of CPUs found. This tells SAS to go and use as many processors as it can access. In some situations, it may be wise to limit the number of CPUs accessed by SAS by setting this value to a maximum CPU number. CPUCOUNT is a way to throttle back the number of processors used in multiple processor environments.

Benchmarking

In threading environments, it is possible that the CPU time used to process may actually be larger than real time/wall time. This may be a consideration when scheduling large production jobs.

Scalable Performance Data Architecture

Since Version 6 of SAS, the Scalable Performance Data Server has been available as a separate product that allows for breaking apart data storage to speed up I/O and avoid the traditional 'serial based' I/O architecture. SPDS Software also supported a very intense security model for access to this data. The ideal storage of a data source would be split across several storage locations that were usually separate disks with separate disk controllers.

In Version 9, this entire architecture, except for the security model, is included.

Scalable Performance Data Engine SPDE

This is a new engine that is part of Base SAS software. It is designed to be used on a libname statement for much quicker access to data stored on disk. The ideal environment is the same as listed above.

Each data location is stored on a separate disk
Each disk has a separate disk controller
Each metadata repository is stored in a separate location
Each index is stored in a separate location

Scalable Analogy

Consider a two lane highway with a two toll booth. Traffic can proceed through those toll booths at a particular speed.

Now consider a two lane highway with 10 toll booths. People split apart to pay the tolls and then merge back together back into a two lane highway.

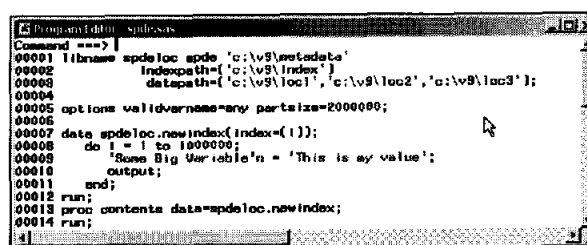
Now consider a 10 lane highway with 10 toll booths.

Now consider a 10 lane highway with no toll booths.

Which one would be fastest?

SPDE performs best when everything is separated into its specific tasks with no stopgaps to slow down processes e.g. a shared disk controller (bad). Performance is best on a 10 lane highway with no toll booths.

Consider the following example to demonstrate the syntax.



```
Command ===>
00001 libname spdeloc spde 'c:\v9\metadata'
00002         indexpath='c:\v9\index'
00003         datapath='c:\v9\loc1', 'c:\v9\loc2', 'c:\v9\loc3';
00004
00005 options validvarname=any partsize=2000000;
00006
00007 data spdeloc.newindex(index=(1));
00008   do i = 1 to 100000;
00009     'Some Big Variable'n = 'This is my value';
00010     output;
00011   end;
00012 run;
00013 proc contents data=spdeloc.newindex;
00014 run;
```

This form of the libname statement uses the SPDE engine.

There is a separate location for:

Metadata

Datapath1
Datapath2
Datapath3
Indexes

This example is for syntax only. In an ideal situation, the locations would all reside on separate disks with different disk controllers.

Conclusion

Large amounts of data can be processed effectively with this architecture. The location of data is now split and must be maintained. This is a consideration when moving data. However, the performance gained may be worth the additional housekeeping.

For additional information on SPDE, see

<http://www.destinycorp.com/documentation/spde/index.html>

PROC CONTENTS

The Contents procedure has been updated to display the native environment of the file, the encoding value and the generation data group.

PROC COPY

The FORCE option has been added to allow MOVEing a data set with audit trails.

The data set attributes are now copied using the CLONE option.

PROC EXPORT

The EXPORT procedure now supports writing to Microsoft Access and Excel 2002 files.

The EXPORT procedure also supports SHEET destinations with the new SHEET= option.

PROC FORMAT

The FORMAT procedure now supports numeric format names that are 32 bytes long and character format names 31 bytes long.

The FORMAT procedure now supports numeric informat names that are 31 bytes long and character informat names 30 bytes long.

PROC FREQ

Multi-threading is not supported for this procedure.

PROC IMPORT

The IMPORT procedure now supports reading to Microsoft Access and Excel 2002 files.

PROC MEANS

This procedure supports multi-threading by default and uses the THREADS/NOTHREADS options.

PROC PRTDEF

This procedure is useful for batch definitions of printers, especially under the UNIX environment.

PROC PRTEXP

This procedure allows for writing attributes of printers to SAS data sets for easy distribution of printer settings.

PROC REGISTRY

The new LISTREG options presents registry information to the SAS Log.

Registries of two different SAS installations can be compared by using the COMPAREREG1 and COMPAREREG2 options.

PROC REPORT

The BEST12. format is now the default numeric format.

This procedure supports multi-threading by default and uses the THREADS/NOTHREADS options.

PROC SORT

The new DATECOPY option uses the original date and time stamp for the new SAS data set.

The new THREADS/NOTHREADS options turns sorting in a multi threaded mode on and off on the procedure level.

PROC SUMMARY

This procedure supports multi-threading by default and uses the THREADS/NOTHREADS options.

PROC SQL

New Dictionary tables are now supported. The complete list is below.

RTF Output

Margin and Indent Options

This option is designed to control margins and indentations in output.

```

Command ==>
00001 options orientation=landscape;
00002 ods pdf file='c:\marginindent.pdf' startpage=no columns=1;
00003 ods rtf file='c:\marginindent.rtf' startpage=no columns=1;
00004 proc report data=evad.bpl need;
00005 column patient comment1 comment2 comment3 comment4;
00006 define patient / group;
00007 define comment1 / computed style={cellwidth=200pt} 'Normal width';
00008
00009 define comment2 / computed
00010 style={leftmargin=12pt} 'LeftMargin Justified';
00011 define comment3 / computed
00012 style={leftmargin=12pt indent=12pt}
00013 'LeftMargin Indented';
00014 define comment4 / computed
00015 style={leftmargin=12pt indent=-12pt}
00016 'LeftMargin reverse(negative) Indent';
00017
00018 compute comment1 / length=50 character;
00019 comment1 = 'The patient's adverse events were moderate. Outcome-';
00020 endcomp;
00021 compute comment2 / length=50 character;
00022 comment2 = 'The patient's adverse events were mild. Outcome-cure';
00023 endcomp;
00024 compute comment3 / length=50 character;
00025 comment3 = 'The patient had no adverse events. Outcome-NH.';
00026 endcomp;
00027 compute comment4 / length=50 character;
00028 comment4 = 'The patient changed dosage. Removed from the study.';
00029 endcomp;
00030 run;
00031 ods _all_ close;

```

PDF Output

RTF Output

Text Options

This option is designed to allow placement of text in any location around ODS output.

PDF Output

Orientation Option

This option allows the changing of output orientation. Notice the placement of the system option.

```

Command ==>
00001 options orientation = landscape;
00002 ods pdf file = 'c:\ods\test.rtf';
00003 proc print data = madselp.vmsa;
00004 run;
00005 options orientation = portrait;
00006 * Notify ODS RTF that the orientation is now portrait;
00007 ods rtf;
00008 proc print data = madselp.vmsa;
00009 run;
00010 ods rtf close;

```

Page X of Y Support

SAS now supports the ability to put page numbers on output with the total pages. This is currently supported for RTF.

```

Command ==>
00001 ods newspaper = 1;
00002 title1 'Page X of Y example.' var 'Page {<page>}';
00003 title2 'This works in RTF only.';
00004 ods rtf file='c:\pageofy.rtf';
00005 proc print data=sashelp.vmsa;
00006 run;
00007 ods rtf close;

```

New ODS Styles

Decimal Alignment

```

Command ==>
00001 data work.decimal;
00002 input @01 charvar $15.
00003 @08 $16 number 0.;
00004 cards;
00005 123 ( 4.8X) 42.6
00006 45 ( 4.8X) 456.99
00007 6 ( 5.7X)
00008 789 ( 1.6X) 88
00009 10 ( 11.4X) 5.0
00010 run;
00011 ods rtf file='decimal.rtf';
00012 title Regular Listing;
00013 proc print data=work.decimal;
00014 run;
00015 title Decimal Aligned Listing;
00016 proc print data=work.decimal;
00017 var charvar number / style(COLUMN)=(just-d)
00018 run;
00019 run;
00020 ods rtf close;

```

125 (8.8%)	42.68
45 (3.1%)	426.90
5 (0.3%)	
739 (51.6%)	88.80
10 (0.7%)	3.80

Cross Environment Data Access (CEDA)

CEDA is still supported in Version 9. CEDA is designed for cross operating system access of data files without the need for PROC UPLOAD/DOWNLOAD or creating a transport file.

In Version 9, the SAS log lists messages if CEDA is being used.

CEDA automatically converts 32 bit and 64 bit data on different operating systems.

Libref Inheritance

Client session defined librefs are now inherited and do not have to be reassigned in multiple server session environments. They are read/write.

%SYSLPUT CONNECTREMOTE= Option

Macro variables can be created on a remote session. Because multiple remote sessions may be available, the CONNECTREMOTE= option allows specification of the session name.

Remote Library Services

Version 9 clients do not connect to Version 6 remote sessions using this feature anymore.

MP CONNECT

This enhances traditional SAS/Connect software with asynchronous processing that is now production. Asynchronous processing is available on remotely submitted servers or processes on multiple processor systems. MP CONNECT processing can increase the performance of processes.

Traditional Example

This is a simple example of how MP CONNECT can work.

```

Command ===>
00001 /* Global SAS system options to start an MP CONNECT remote session. */
00002 option autosign=yes;
00003 option sasacid="sas";
00004
00005 /* Remote submit first task. */
00006 rsubmit process=task1 wait=no;
00007
00008 libname temp 'c:\vs';
00009 data work.class;
00010 do i = 1 to 100000;
00011 name = 'BAG';
00012 output;
00013 end;
00014 run;
00015 proc sort data=work.class out=temp.class1;
00016 by descending i;
00017 run;
00018
00019 endrsubmit;
00020
00021 /* Remote submit second task. */
00022 rsubmit process=task2 wait=no;
00023
00024 libname temp 'c:\vs';
00025 data work.class;
00026 do i = 1 to 100000;
00027 name = 'BAG';
00028 output;
00029 end;
00030 run;
00031 proc sort data=work.class out=temp.class2;
00032 by descending i;
00033 run;
00034
00035 endrsubmit;
00036
00037 /* Wait for both tasks to complete. */
00038 waitfor _all_ task1 task2;
00039

```

```

Command ===>
00040 /* Get the remote logs and place them in the current log */
00041 rget task1;
00042 rget task2;
00043
00044 /* Merge the results and continue processing. */
00045 libname temp 'c:\vs';
00046 data work.sorted;
00047 merge temp.class1 temp.class2;
00048 run;

```

```

SAS LOG (Task 1)
Command ===>
01 /* Global SAS system options to start an MP CONNECT remote session. */
02 option autosign=yes;
03 option sasacid="sas";
04
05 /* Remote submit first task. */
06 rsubmit process=task1 wait=no;
07
08 libname temp 'c:\vs';
09 data work.class;
10 do i = 1 to 100000;
11 name = 'BAG';
12 output;
13 end;
14 run;
15 proc sort data=work.class out=temp.class1;
16 by descending i;
17 run;
18
19 endrsubmit;
20
21 /* Remote submit second task. */
22 rsubmit process=task2 wait=no;
23
24 libname temp 'c:\vs';
25 data work.class;
26 do i = 1 to 100000;
27 name = 'BAG';
28 output;
29 end;
30 run;
31 proc sort data=work.class out=temp.class2;
32 by descending i;
33 run;
34
35 endrsubmit;
36
37 /* Wait for both tasks to complete. */
38 waitfor _all_ task1 task2;
39
40 rget task1;
41 rget task2;
42
43 /* Merge the results and continue processing. */
44 libname temp 'c:\vs';
45 data work.sorted;
46 merge temp.class1 temp.class2;
47 run;

```

NOTE: SAS initialization used:
real time 1.54 seconds
cpu time 0.02 seconds

NOTE: Remote signon to TASK1 complete.
NOTE: Background remote submit to TASK1 in progress.

01 /* Remote submit first task. */
02 rsubmit process=task1 wait=no;
03
04 libname temp 'c:\vs';
05 data work.class;
06 do i = 1 to 100000;
07 name = 'BAG';
08 output;
09 end;
10 run;
11 proc sort data=work.class out=temp.class1;
12 by descending i;
13 run;
14
15 endrsubmit;
16
17 /* Remote submit second task. */
18 rsubmit process=task2 wait=no;
19
20 libname temp 'c:\vs';
21 data work.class;
22 do i = 1 to 100000;
23 name = 'BAG';
24 output;
25 end;
26 run;
27 proc sort data=work.class out=temp.class2;
28 by descending i;
29 run;
30
31 endrsubmit;
32
33 /* Wait for both tasks to complete. */
34 waitfor _all_ task1 task2;
35
36 rget task1;
37 rget task2;
38
39 /* Merge the results and continue processing. */
40 libname temp 'c:\vs';
41 data work.sorted;
42 merge temp.class1 temp.class2;
43 run;

NOTE: SAS initialization used:
real time 1.54 seconds
cpu time 0.02 seconds

NOTE: Remote signon to TASK2 complete.
NOTE: Background remote submit to TASK2 in progress.

01 /* Remote submit first task. */
02 rsubmit process=task1 wait=no;
03
04 libname temp 'c:\vs';
05 data work.class;
06 do i = 1 to 100000;
07 name = 'BAG';
08 output;
09 end;
10 run;
11 proc sort data=work.class out=temp.class1;
12 by descending i;
13 run;
14
15 endrsubmit;
16
17 /* Remote submit second task. */
18 rsubmit process=task2 wait=no;
19
20 libname temp 'c:\vs';
21 data work.class;
22 do i = 1 to 100000;
23 name = 'BAG';
24 output;
25 end;
26 run;
27 proc sort data=work.class out=temp.class2;
28 by descending i;
29 run;
30
31 endrsubmit;
32
33 /* Wait for both tasks to complete. */
34 waitfor _all_ task1 task2;
35
36 rget task1;
37 rget task2;
38
39 /* Merge the results and continue processing. */
40 libname temp 'c:\vs';
41 data work.sorted;
42 merge temp.class1 temp.class2;
43 run;

Notice the RGET statements to allow the local SAS log to see the remote SAS log information.

Notice the WAITFOR statements which must be used at strategic points in processing.

Piping Example

Piping is a new methodology in SAS where the output of one step is automatically being fed to the input of a subsequent step. When the process is appropriate, for example, the creation of a data set with a data step feeding directly into a subsequent proc sort, performance increases are possible. The output of the first process is not written to disk. The input of the second process does not read from disk. All information is passed through memory.

This is a simple example of how piping might be used. Notice the use of real physical storage to pass data between sessions.

```

Command ===>
00001 /* Global SAS system options to start an HP CONNECT remote session. */
00002 libname temp c:\v8;
00003 option autosign=yes;
00004 option sasacd="sas";
00005 signon task1 sasacd="sasacd";
00006 signon task2 sasacd="sasacd";
00007 /* Remote submit first task. */
00008 submit process=task1 wait=no;
00009 libname pipe1 sassock "pipe1";
00010 data pipe1.class1;
00011 do i = 1 to 1000000;
00012 name = 'BNS';
00013 output;
00014 end;
00015 run;
00016 endsubmit;
00017
00018 libname pipe1 sassock "pipe1";
00019 proc sort data=pipe1.class1 out=temp.class1;
00020 by descending i;
00021 run;
00022
00023 submit process=task2 wait=no;
00024 libname pipe2 sassock "pipe2";
00025 data pipe2.class2;
00026 do i = 1 to 1000000;
00027 name = 'BNS';
00028 output;
00029 end;
00030 run;
00031 endsubmit;
00032
00033 libname pipe2 sassock "pipe2";
00034 proc sort data=pipe2.class2 out=temp.class2;
00035 by descending i;
00036 run;
00037 /* Wait for both tasks to complete. */
00038 waitfor _all_ task1 task2;
00039
00040 /* Get the remote logs and place them in the current log */
00041 rget task1;
00042 rget task2;
00043
00044 /* Merge the results and continue processing. */
00045 data work.sorted;
00046 merge temp.class1 temp.class2;
00047 run;

```

In addition, the pipe locations must be listed in the services file. Notice pipe1 – pipe4 in the file listed below.

File	Edt	Format	Help
radacct		1813/udp	#RADIUS accounting
protocol			
nfsd		2049/udp	#NFS server
knetd		2053/tcp	#Kerberos
de-multiplexor			
man		9333/tcp	#Remote Man Server
#AppDevstudio service			
shr1		5010/tcp	#SAS/SHARE SERVER
pipe1		5020/tcp	#pipe1
pipe2		5021/tcp	#pipe2
pipe3		5022/tcp	#pipe3
pipe4		5023/tcp	#pipe4

New Graph Procedures

Three new procedures now exist in SAS/GRAPH software.

GBARLINE

This procedure creates vertical bar charts with a plot line.

GAREABAR

This procedure creates bar charts where the width of the bar represents a prescribed value.

MAPIMPORT

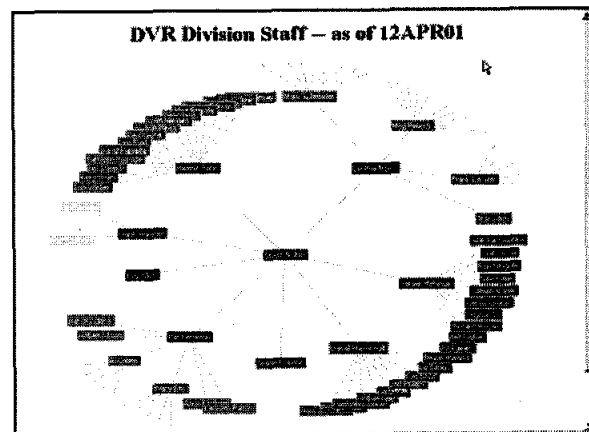
This procedure imports map information.

Java Applets Supported by SAS/Graph

Treeview Java Applet

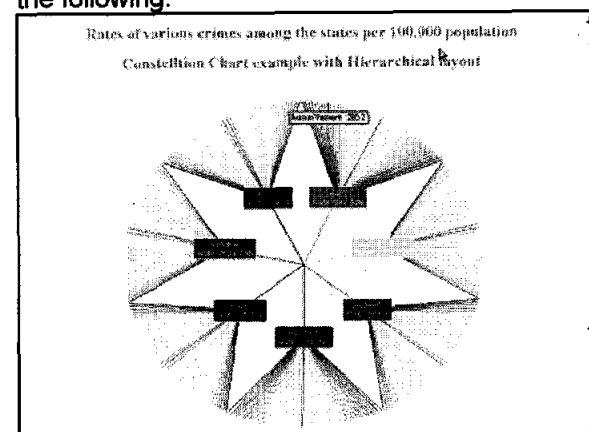
The new Treeview macro has been introduced to create a Java Applet for an interactive 'tree' style report. This creates a JAR file that is publishable via the web. The following code is a snapshot of the beginning of the macro.

This macro and all subsequent macros can be found in the following directory (under Windows). In a web browser, the resulting JAR file yields the following, interactive display.



Constellation Java Applet

Another macro exists to create a constellation JAR file. The result of running this macro yields an interactive JAR file like the following.



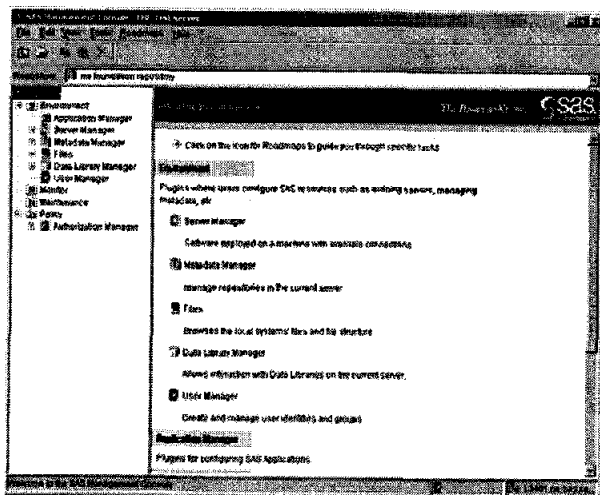
SAS Management Console

The SAS Management Console is designed to allow for managing and monitoring metadata, servers, libraries of data, servers and security from one, central point in an organization.

This application also supports third part plug ins.

Aspects of the management console include:

- Server Manager – manage SAS servers
- Metadata Manager –interact with running repository servers
- Application Manager – SAS application plug-ins
- SAS Library Manager – define and manage SAS libraries
- User Manager – define SAS Users and Groups
- Authorization Manager - administer authorization policy for accessing SAS Metadata and OLAP



The benefits of the SAS Management Console are:

- Simplify administrative tasks by using the same tools for all SAS products and solutions.
- Reduce staff training and support time.
- Build standard and repeatable processes for SAS operations.
- Define and manage connections to servers:
- Application
- Database
- SAS
- IOM Bridge
- Others
- Define and manage SAS users and groups.
- Policy descriptions: set values for user and group attributes (such as read, write, etc.)
- Manage SAS Library definitions.
- Manage Database Schemas.
- Administer authorization policy for accessing SAS Metadata and OLAP:
- Permission Creation
- Access Control Templates
- Resource Authorization Definitions
- Supports plug-ins to administer SAS applications, or user-written SAS applications.
- Manage SAS licenses (SETINIT):
- The SAS License Manager (SLM) displays SAS installed software information.
- Monitor SAS processes:
- Determine where SAS is running
- Identify user of a SAS process
- Identify "orphans"
- Interface with resource managers

Operating Systems Supported

SAS V9 is supported under all of the Windows operating systems, including NT, 2000 and XP, but not supported under Windows 95, 98 and Me.

Operating System	Size
Windows NT4.0/2000/XP (WNT/W2K/WXP)	32 bit
OpenVMS Alpha 7.2	64 bit
Compaq's Digital UNIX 5.1	64 bit
HP HP-UX 64bit 11.0	64 bit
Solaris 64bit Solaris8	64 bit
AIX 64bit 5.1	64 bit
RedHat Linux 7.2 on Intel	32 bit
OS/390 (MVS) V2R10	32 bit

Conclusion

As you can see, SAS Version 9 offers a whole new way of processing that yields more choices to take your programming to the next level.