



## CHAPTER

## 3

# Accessing SAS Data Libraries

<i>Introduction</i>	<b>51</b>
<i>Utilities That You Can Use with Direct Access Bound Libraries</i>	<b>53</b>
<i>SAS Data Sets in HFS Libraries</i>	<b>53</b>
<i>File Sharing in HFS Libraries</i>	<b>54</b>
<i>Using Version 7 Engines</i>	<b>54</b>
<i>Using Version 6 and Version 5 Engines</i>	<b>55</b>
<i>Using V8 Engines</i>	<b>55</b>
<i>When to Use This Engine</i>	<b>55</b>
<i>How to Select the V8 Engine</i>	<b>56</b>
<i>DCB Attributes for Direct Access Bound Libraries with the V8 Engine</i>	<b>56</b>
<i>External Allocation: V8 Engine</i>	<b>56</b>
<i>Internal Allocation: V8 Engine</i>	<b>57</b>
<i>Engine/Host Options for the V8 Engine</i>	<b>57</b>
<i>CONTENTS Procedure Output</i>	<b>58</b>
<i>Using the V8TAPE Engine</i>	<b>59</b>
<i>When to Use This Engine</i>	<b>59</b>
<i>How to Select the V8TAPE Engine</i>	<b>60</b>
<i>DCB Attributes for Direct Access Bound Libraries with the V8TAPE Engine</i>	<b>60</b>
<i>External Allocation: V8TAPE Engine</i>	<b>60</b>
<i>Internal Allocation: V8TAPE Engine</i>	<b>61</b>
<i>Engine/Host Options for Direct Access Bound Libraries with the V8TAPE Engine</i>	<b>61</b>
<i>CONTENTS Procedure Output</i>	<b>62</b>

## Introduction

The V8 and V8TAPE engines access V8 and V8TAPE SAS data libraries, respectively. Table 3.1 on page 52 summarizes useful information about these engines. The V8 engine supports direct access to individual members of a library, without having to search through the entire library sequentially. The V8TAPE engine supports sequential format data libraries on tape or on disk.

In SAS Version 8, a SAS library can exist as one of the following:

### Direct Access Bound Library

A SAS data library in a direct access engine format (V8 or V6 engine) that is implemented with a single specially formatted OS/390 disk data set.

### Sequential Access Bound Library

A SAS data library which exists in a sequential engine format (V8TAPE or V6TAPE), implemented within a single OS/390 disk or tape data set.

**HFS Library**

Is a collection of SAS files with the same engine format (V8 or V8TAPE) that are stored in a directory of the hierarchical file system of UNIX System Services.

**Hiperspace Library**

Is a collection of SAS files with the same engine format that are loaded into an OS/390 hiperspace, which is virtual storage backed by available expanded storage.

For general information about SAS library engines, see “SAS Library Engines” on page 32 and *SAS Language Reference: Dictionary*.

**Table 3.1** Overview of the V8 and V8TAPE Engines

	<b>V8 Engine</b>	<b>V8TAPE Engine</b>
Member Types Supported	ACCESS	ACCESS*
	CATALOG	CATALOG*
	DATA**	DATA
	ISTORE	ISTORE
	PROGRAM	PROGRAM*
	VIEW	VIEW*
OS/390-Specific Options for Bound Libraries (specified in LIBNAME statement)	BLKSIZE=	BLKSIZE=
	DATACLAS=	DATACLAS=
	DISP=	DISP=
	DLTRUNCCHK	n/a
	EXTEND	EXTEND
	HIPERSPACE	n/a
	LABEL=	LABEL=
	LIKE=	LIKE=
	LINEAR	LINEAR
	MGMTCLASS=	MGMTCLASS=
	PROMPT	PROMPT
	SPACE=	SPACE=
	STORCLAS=	STORCLAS=
	UNIT=	UNIT=
	VOLSER=	VOLSER=
	WAIT=	WAIT=
OS-390-Specific Options for HFS Libraries (specified In LIBNAME statement)	HFS	HFS
	PROMPT	PROMPT
Portable Data Set Options	All	All except for COMPRESS=, INDEX=, and REUSE=
OS/390 Specific Data Set Options	n/a	FILEDISP=

OS/390-Specific System Options	BLKSIZE=	FILEDEV=
	BLKSIZE0=	FILESPPRI=
	FILEDEV	FILESPEC=
	FILESPPRI=	FILEUNIT=
	FILESPEC=	FILEVOL=
	FILEUNIT=	SYSPREF=
	FILEVOL=	TAPECLOSE=
	SYSPREF=	
DCB Attributes for Bound Libraries	DSORG=PS	DSORG=PS
	RECFM=FS	RECFM=U
	BLKSIZE=<value>, where <value> is in the range 4096-32,256 in increments of 512. The default is 6144.	BLKSIZE=<value>, where <value> is up to the maximum for your device type.
	LRECL=<value>, same as BLKSIZE=	LRECL=any value

\* Limited support: you can use the V8TAPE engine to move or transport these member types, but you cannot use the V8TAPE engine to access the information within these members.

\*\* Including indexing and audit trail capabilities.

---

## Utilities That You Can Use with Direct Access Bound Libraries

You can use standard OS/390 utilities such as IEBGENER, ISPF/PDF 3.3, or DF/HSM to copy or move SAS data libraries that were created with the V8 engine or with the V8TAPE engine. As long as the block size of the data library that you are copying is not changed and not greater than the track capacity of the target device, a V8 or V8TAPE data library may be transferred between unlike device types (for example, 3350 to 3380). This makes it easier to use DASD management software such as IBM's DF/HSM, FDR by Innovation Data Processing, or DMS/OS by Sterling Software.

---

## SAS Data Sets in HFS Libraries

SAS data sets in HFS libraries are stored as individual HFS files within the directory that defines the library. The filename format is lowercase, with a period and filename extension provided by SAS.

### **CAUTION:**

**Do not change the filename extensions of SAS files.** Filename extensions determine how SAS accesses files; changing them can cause unpredictable results. △

The following table describes SAS filename extensions.

**Table 3.2** File Extensions for SAS Files in HFS Libraries

Random Access Files	Sequential Access Files	SAS Member Type	Description
.sas7bdat	.sas7sdat	DATA	SAS data file
.sas7bndx	.sas7sndx	INDEX	Data file index; not treated by the SAS System as a separate file
.sas7bcatalog	.sas7scat	CATALOG	SAS catalog
.sas7bpgrm	.sas7spgrm	PROGRAM	Stored program (DATA step)
.sas7bview	.sas7sview	VIEW	SAS data view
.sas7bacs	.sas7sacs	ACCESS	Access descriptor file
.sas7baud	.sas7saud	AUDIT	Audit file
.sas7bfdb	.sas7sfdb	FDB	Consolidation database
.sas7bmdb	.sas7smdb	MDDb	Multidimensional database
.sas7bods	.sas7sods	SASODS	Output delivery system file
.sas7bdmd	.sas7sdmd	DMDB	Data mining database
.sas7bitm	.sas7ssitm	ITEMSTOR	Item store file
.sas7butl	.sas7sutl	UTILITY	Utility file
.sas7bput	.sas7sput	PUTILITY	Permanent utility file
.sas7bbak	.sas7sbak	BACKUP	Backup file

### File Sharing in HFS Libraries

When SAS creates or updates a member of an HFS library, it places an exclusive lock on the entire file. The lock prevents other jobs, processes, or SAS sessions from reading, writing, or updating that file until the lock is removed. The write lock is analogous to a DISP=OLD data set enqueue.

When SAS reads an existing member of an HFS library, it places a read (or shared) lock on the entire file, which prevents other jobs, processes, or SAS sessions from updating the file, though it is still possible for others to read the file. The read lock is analogous to DISP=SHR data set enqueueing.

## Using Version 7 Engines

The V7 engine is identical to the V8 engine. SAS identifies V7 libraries and members thereof as having the V8 engine format. Members created with the V7 and V8 engines may coexist in the same library. This information also applies to the V7TAPE and V8TAPE engines.

---

## Using Version 6 and Version 5 Engines

The V6 and V6TAPE engines supplied with SAS Version 8 support full access (read, write, and create) to all SAS Version 6 data libraries. The V6 and V6TAPE engines take the same options as the V8 and V8TAPE engines.

Note that data sets created under SAS Release 6.06 cannot be processed by SAS Version 8 because their storage format differs from that used in subsequent releases of SAS Version 6. To make a SAS 6.06 data set available for processing in Version 8, first use a later release of SAS Version 6 (6.07, 6.08, or 6.09) to copy the 6.06 data set to a new data set, either in the same library or in a new library. You can copy the data set into a 6.06 library because all library formats are supported in Version 8. The newly copied data set automatically receives the new Version 6 format, which allows the new data set to be processed by the V6 or V6TAPE engine in SAS Version 8.

The V5 and V5TAPE engines in SAS Version 8 support read-only access to SAS Version 5 data libraries. For information on V5 and V5TAPE engines, see Appendix 1 of *SAS Companion for the MVS Environment*.

For information about the portable features of the SAS that are supported by these engines, see *SAS Language Reference: Dictionary*.

---

## Using V8 Engines

A V8 data library is a physical file that is formatted to be compatible with the V8 engine. The V8 engine format enables you to access any of the SAS data sets in the library directly, without searching through other SAS data sets. You can read or write more than one SAS data set in a disk-format data library in a single DATA step. Note that preceding does not apply to SAS HFS data libraries stored in UNIX System Services directories.

---

## When to Use This Engine

You use the V8 engine to create SAS data libraries on disk and to read from, write to, or update those libraries. The V8 engine also enables you to index and compress observations. For more information about indexes, see *SAS Language Reference: Dictionary*. For more information about compressing observations, see *SAS Language Reference: Dictionary* and “Determine whether you should compress your data” on page 152.

The V8 engine is the default engine for SAS data libraries, unless the default engine has been changed with the SAS system option `ENGINE=`. This engine is also the only engine under OS/390 that provides full support for catalogs in SAS Version 8; members of type ACCESS, PROGRAM, and VIEW; and SAS data libraries under UNIX System Services. It also supports multivolume SAS data libraries on disk. For further information on multivolume libraries, see “Allocating Multivolume SAS Data Libraries” on page 41.

Here are some other important characteristics of the V8 engine for direct access bound data libraries:

- The V8 engine makes efficient use of DASD space because it uses fixed-length physical blocks and enables you to have more than one SAS file per track.
- You can use standard utilities to move or copy a V8 data library between unlike device types. This can be done only if the block size of the library does not change and is less than or equal to the track size of the target device.

Here are additional characteristics of the V8 engine that apply to all library implementations (direct access bound library, HFS library, and hiperspace library):

- The V8 engine provides a convenient way for you to reduce the elapsed time for SAS jobs. Setting the BUFNO= system option or data set option to a value greater than 1 causes multiple pages to be transferred during a single DASD I/O operation. This can substantially reduce the elapsed time for SAS jobs. However, multiple pages may not necessarily be transferred when an index is used in a “read” operation.
- It exploits memory above the 16-megabyte line. I/O buffers for permanent V8 data libraries are allocated above the 16-megabyte line.
- After a V8 data library is opened, it remains open until the LIBREF is cleared, or until you end the SAS session. At step boundaries within a SAS session, SAS ensures data integrity by writing any necessary control information to the data set.

---

## How to Select the V8 Engine

There are four ways to select the V8 engine:

- Specify V8 as the value of the *engine* argument in the LIBNAME statement or LIBNAME function.

*Note:* Use BASE as the engine name if you write programs that create new SAS data libraries and you want to create the data libraries in the latest available format. In Version 8, BASE is an alias for V8, and it will be an alias for newer engines in subsequent releases. △

- For existing V8 direct access bound data libraries, specify no value for *engine* in the LIBNAME statement or LIBNAME function. SAS then examines the data set attributes and selects the V8 engine automatically. SAS also selects the V8 engine automatically if you omit the LIBNAME statement or function and use a JCL DD statement or a TSO ALLOCATE command to allocate the library.
- Set the value of the SAS system option ENGINE= to V8. This option tells SAS which engine to use as the default when you allocate a new physical file without specifying an engine.
- Specify V8 in the Engine field in the SAS Explorer New Library dialog box.

---

## DCB Attributes for Direct Access Bound Libraries with the V8 Engine

The physical file label contains DCB information that describes the data set's characteristics. The operating environment writes the DCB information when it creates the library, using either values that are supplied by the user or the values of several SAS system options. Both the SAS System and OS/390 utility programs use this DCB information during processing.

The following sections provide additional information about DCB parameters for the V8 engine. Also see “DCB Attribute Options” on page 300 for more information.

### External Allocation: V8 Engine

If you use a JCL DD statement or a TSO ALLOCATE command to allocate a new V8 SAS data library, and if you choose to specify DCB attributes, then you must specify the following DCB attributes:

- DSORG=PS
- RECFM=FS

- BLKSIZE=4096 to 32256 in increments of 512 (The default is 6144.)
- LRECL=*value*, where *value* is the same as the value for BLKSIZE=.

If you do not specify DCB attributes for a new V8 SAS data library, then SAS supplies the above DCB attributes whenever the V8 engine has been either explicitly or implicitly assigned. (See “How to Select the V8 Engine” on page 56 and “How SAS Assigns an Engine When No Engine Is Specified” on page 40 for details about how engines are assigned.)

When allocating a V8 data library, choose an appropriate block size. Here are some guidelines. (See also the performance information in “Optimizing I/O” on page 151.)

- If most members of the library will have large numbers of observations, choose the optimal block size for the device. (For a 3380, this value is 23040. For a 3390, it is 27648. See your IBM documentation for the optimal block sizes of other devices.) The result is the minimum amount of overhead that is required to map and transfer library DASD blocks, because there will be fewer total blocks to handle.
- If most members of the library will contain few observations, or if their size and number of observations vary widely, choose a smaller block size such as 6144. This increases the opportunity for apportioning DASD space among library members, and it generally makes better use of DASD space.

If no block size is specified when a SAS data library is physically opened for the first time, SAS uses the value of the BLKSIZE= system option. If that value is 0, then SAS uses the BLKSIZE(*device-type*)= system option. If the value of BLKSIZE(*device-type*)= is 0, SAS uses a block size of 6144 by default.

If you specify a block size that is not an integral multiple of 512, then the number is rounded down to an integral multiple of 512.

## Internal Allocation: V8 Engine

If you use the LIBNAME statement or LIBNAME function to allocate a new V8 SAS data library, SAS supplies the following DCB attributes for you:

- DSORG=PS
- RECFM=FS
- BLKSIZE=6144
- LRECL=6144

You can override the default value of the BLKSIZE= system option by using the BLKSIZE= option in the LIBNAME statement or LIBNAME function.

*Note:* If the SAS system option BLKALLOC is in effect, then SAS assigns BLKSIZE= and LRECL= values when the SAS data library is allocated. If NOBLKALLOC is in effect, then SAS assigns BLKSIZE= and LRECL= values when the library is first accessed. △

---

## Engine/Host Options for the V8 Engine

The engine/host options that you can supply in the LIBNAME statement, LIBNAME function, or SAS Explorer New Library Assignment dialog box correspond to the JCL or TSO parameters that you would specify if you allocated the SAS data library externally. For bound libraries using the V8 engine, you can specify any of the engine/host options shown in Table 3.1 on page 52. (For more information about these options and their values, see the description of engine/host options under “LIBNAME” on page 313.) Or you can accept the default values that are derived from the corresponding SAS system options, as follows:

- If you do not specify a value for DISP=, the default for existing data sets is DISP=(OLD,KEEP,KEEP). For new data sets, the default depends on how you are allocating the library:
  - If you are allocating the library with a LIBNAME statement, LIBNAME function, or with a TSO ALLOCATE command, disposition defaults to DISP=(NEW,CATLG,DELETE).
  - If you are allocating the library with a JCL DD statement, disposition defaults to DISP=(NEW,DELETE,DELETE).
  - For HFS libraries, the DISP= option is ignored.

In an interactive environment, if you use a LIBNAME statement, LIBNAME function, or the SAS Explorer New Library Assignment dialog box to allocate a data library that does not exist and you do not specify a value for DISP=, one of the following actions occurs:

- If the SAS system option FILEPROMPT is in effect (the default), then a requestor window asks whether you want to create the physical file. If you reply **yes**, you are asked whether you want to catalog the data set when it is deallocated, or to delete it.
 

For HFS libraries, a requestor window asks whether you want to create the file, but you do not get a second requestor window.
- If the SAS system option NOFILEPROMPT is in effect, an error message is written to the SAS log.
- If you do not specify values for the SPACE= parameters, SAS uses the current values of the SAS system options FILEUNIT=, FILESPPRI=, and FILESPSEC=. The defaults are SPACE=(CYL,(1,1)).
- If you do not specify a value for VOLSER=, SAS uses the current value of the SAS system option FILEVOL=, if a value for FILEVOL= has been specified at your site.
- If you do not specify a value for UNIT=, SAS uses the current value of the SAS system option FILEDEV=. The default is SYSDA.

*Note:* The default values shown are those that are supplied by SAS. Your SAS system administrator may have changed the default values for your site.  $\Delta$

For temporary data libraries, you do not need to specify any options, but you can override any of the default values.

---

## CONTENTS Procedure Output

The PROC CONTENTS output in Output 3.1 on page 58 shows information that is generated by the V8 engine.



**Output 3.1** PROC CONTENTS Output Generated by the V8 Engine

```

The SAS System
The CONTENTS Procedure
1

Data Set Name: WORK.ORANGES      Observations: 4
Member Type: DATA                Variables: 5
Engine: V8                        Indexes: 0
Created: 14:27 Tuesday, March 5, 1999 Observation Length: 40
Last Modified: 14:27 Tuesday, March 5, 1999 Deleted Observations: 0
Protection:                      Compressed: NO
Data Set Type:                   Sorted: NO
Label:

-----Engine/Host Dependent Information-----

Data Set Page Size: 6144
Number of Data Set Pages: 1
First Data Page: 1
Max Obs per Page: 152
Obs in First Data Page: 4
Number of Data Set Repairs: 0
Physical Name: SYS96065.T142625.RA000.USERID.R0000180
Release Created: 7.0000B2
Release Last Modified: 7.0000B2
Created by: USERID
Last Modified by: USERID
Subextents: 1
Total Blocks Used: 1

-----Alphabetic List of Variables and Attributes-----

# Variable Type Len Pos Format Informat Label
-----
2 FLAVOR Num 8 8
4 LOOKS Num 8 24
3 TEXTURE Num 8 16
5 TOTAL Num 8 32
1 VARIETY Char 8 0

```

---

## Using the V8TAPE Engine

The V6TAPE engine is functionally equivalent to the V8TAPE engine. Information in this section that applies to V8TAPE also applies to V6TAPE.

---

### When to Use This Engine

Use the V8TAPE engine to create sequential-format SAS data libraries either on disk or on tape, and to access files in sequential data libraries. The primary purpose of this engine is to enable you to back up Version 8 SAS data sets, catalogs, or whole data libraries. The V8TAPE engine makes it possible to back up applications that contain both SAS data sets and SAS catalogs. The V8TAPE engine also enables you to transport a tape-format library to another OS/390 or CMS system via tape or shared DASD. Finally, you can use this engine to create a multivolume sequential-format SAS data library on tape or disk.

In contrast to the V8 engine, V8TAPE has the following limitations:

- It does not support indexing nor compression of observations.
- Because a V8TAPE SAS data library does not contain a directory, SAS cannot access an individual data set directly. It must read through all preceding SAS data

sets in order to reach a requested data set. Direct access to individual observations (using the POINT= or KEY= options in the SET or MODIFY statements) also is not supported.

- ☐ In a single DATA step or PROC step, you can use only one SAS data set from a particular sequential SAS data library.

Unlike disk-format libraries, sequential libraries are always closed at step boundaries.

*Note:* Information in this section also applies to the V6TAPE engine.  $\triangle$

---

## How to Select the V8TAPE Engine

There are three ways to select this engine:

- ☐ Specify V8TAPE as the value of the *engine* argument in the LIBNAME statement or LIBNAME function.

*Note:* Use TAPE as the engine name if you write programs that create new SAS data libraries and you want to create the data libraries in the latest available format. In SAS Version 8, TAPE is an alias for V8TAPE, and it will be an alias for newer sequential engines in subsequent releases.  $\triangle$

- ☐ For existing V8TAPE SAS data libraries on tape, specify no value for *engine* in the LIBNAME statement or LIBNAME function. This engine is selected automatically if you do not provide a value for the engine argument and you are allocating the library internally. This engine is also selected automatically if you omit the LIBNAME statement or function and allocate the library externally.
- ☐ Set the value of the SEQENGINE= system option to SASV8SEQ. The DCB attributes of the allocated data set must indicate a sequential format SAS data library.

---

## DCB Attributes for Direct Access Bound Libraries with the V8TAPE Engine

The physical file label contains DCB information which describes the data library's characteristics. The operating environment writes the DCB information when it creates the library, using either values that are supplied by the user, or the values of several SAS system options. Both the SAS System and OS/390 utility programs use this DCB information during processing.

See "DCB Attribute Options" on page 300 for more information about DCB attributes.

## External Allocation: V8TAPE Engine

If you use a JCL DD statement or a TSO ALLOCATE command to allocate a new V8TAPE data library on disk or on tape, and if you choose to specify DCB attributes, then you must specify the following DCB attributes:

- ☐ DSORG=PS
- ☐ RECFM=U
- ☐ BLKSIZE=*value*, where *value* is up to the maximum for your device type.

If you do not specify DCB attributes for a new V8TAPE SAS data library, then SAS supplies the above DCB attributes whenever the V8TAPE engine has been either explicitly or implicitly assigned. (See "How to Select the V8TAPE Engine" on page 60 and "How SAS Assigns an Engine When No Engine Is Specified" on page 40 for details about how engines are assigned.)

*Note:* In order to use the TSO ALLOCATE command or LIBNAME statement to allocate a data set on tape, your user ID must have MOUNT authority. See your system administrator for details. △

## Internal Allocation: V8TAPE Engine

If you use the LIBNAME statement or LIBNAME function to allocate a new V8TAPE data library on disk or on tape, SAS supplies the following DCB attributes for you:

- DSORG=PS
- RECFM=U

In addition, if no block size is specified when a SAS data library is physically opened for the first time, SAS uses the value of the BLKSIZE= system option. If that value is 0, then SAS uses the BLKSIZE(*device-type*)= system option. If the value of BLKSIZE(*device-type*)= is 0, SAS uses the largest block size that the device supports.

*Note:* If the SAS system option BLKALLOC is in effect, then SAS assigns BLKSIZE= and LRECL= values when the SAS data library is allocated. If NOBLKALLOC is in effect, then SAS assigns BLKSIZE= and LRECL= values when the library is first accessed. △

---

## Engine/Host Options for Direct Access Bound Libraries with the V8TAPE Engine

The engine/host options that you can supply in the LIBNAME statement or LIBNAME function correspond to the JCL or TSO parameters that you would specify if you allocated the SAS data library externally. For bound libraries using the V8TAPE engine, you can specify any of the engine/host options shown in Table 3.1 on page 52. (For more information about these options and their values, see the description of *engine/host options* under “LIBNAME” on page 313.) Or you can accept the default values that are derived from the corresponding SAS system options, as follows:

- If you do not specify a value for DISP=, the default for existing data sets is DISP=(OLD,KEEP,KEEP). For new data sets, the default is DISP=(NEW,CATLG,DELETE). If you specify only DISP=NEW and omit the normal disposition parameter, the following defaults occur:
  - If you are allocating the library with a LIBNAME statement, a LIBNAME function, or with a TSO ALLOCATE command, normal disposition defaults to CATLG.
  - If you are allocating the library with a JCL DD statement, normal disposition defaults to DELETE for new data sets.

In an interactive environment, when you use a LIBNAME statement or LIBNAME function to allocate a data library that does not exist and you do not specify a value for DISP=, one of the following actions occurs:

- If the SAS system option FILEPROMPT is in effect (the default), then a requestor window asks whether you want to create the physical file. If you reply **yes**, you are asked whether you want to catalog the data set when it is deallocated, or to delete it.
- If the SAS system option NOFILEPROMPT is in effect, an error message is written to the SAS log.
- If you do not specify values for the SPACE= parameters, SAS uses the current values of the SAS system options FILEUNIT=, FILESPPRI=, and FILESPSEC=. The defaults are SPACE=(CYL,(1,1)).

- If you do not specify a value for VOLSER=, SAS uses the current value of the SAS system option FILEVOL=, if a value for FILEVOL= has been specified at your site.
- If you do not specify a value for UNIT=, SAS uses the current value of the SAS system option FILEDEV=. The default is SYSDA.

*Note:* The default values shown are those that are supplied by SAS. Your SAS system administrator may have changed the default values for your site.  $\Delta$

For temporary data libraries, you do not need to specify any options, but you can override any of the default values.

---

## CONTENTS Procedure Output

The PROC CONTENTS output in Output 3.2 on page 62 shows information that is generated by the V8TAPE engine.

**Output 3.2** PROC CONTENTS Output Generated by the V8TAPE Engine

```

The SAS System
The CONTENTS Procedure

Data Set Name: SEQ.ORANGES      Observations:      .
Member Type:   DATA            Variables:          5
Engine:        V8TAPE           Indexes:            0
Created:       14:48 Tuesday, March 5, 1999  Observation Length: 40
Last Modified: .                Deleted Observations: 0
Protection:                               Compressed:      NO
Data Set Type:                               Sorted:         NO
Label:

-----Engine/Host Dependent Information-----

Data Set Page Size: 32000
Physical Name:      USERID.OS390.SEQ
Release Created:    8.0
Created by:         USERID

-----Alphabetic List of Variables and Attributes-----

#    Variable    Type    Len    Pos
-----
2    FLAVOR      Num      8      8
4    LOOKS       Num      8     24
3    TEXTURE     Num      8     16
5    TOTAL       Num      8     32
1    VARIETY     Char      8      0

```

In the output example above, SAS version information appears in the Engine and Release created fields.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Companion for the OS/390 Environment, Version 8*, Cary, NC: SAS Institute Inc., 1999.

**SAS® Companion for the OS/390® Environment, Version 8**

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

1-58025-523-X

All rights reserved. Produced in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

**U.S. Government Restricted Rights Notice.** Use, duplication, or disclosure of the software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, November 1999

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

DB2®, IBM®, and OS/2® are registered trademarks or trademarks of International Business Machines Corporation. ORACLE® is a registered trademark or trademark of Oracle Corporation. ® indicates USA registration.

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

The Institute is a private company devoted to the support and further development of its software and related services.