



APPENDIX

2

Advanced User Topics

<i>Introduction</i>	117
<i>Data Set Options</i>	117
<i>DDBKEY= Data Set Option</i>	118
<i>DDBLOAD= Data Set Option</i>	118
<i>DDBLOCK= Data Set Option</i>	118
<i>DDBPW= Data Set Option</i>	118
<i>DDBSV= Data Set Option</i>	119
<i>DDBTRACE= Data Set Option</i>	119
<i>DDBURT= Data Set Option</i>	119
<i>DDBUSER= Data Set Option</i>	119
<i>Using Multiple View Descriptors</i>	120
<i>User Exits</i>	120
<i>Deletions and Insertions</i>	120
<i>Missing Values (Nils)</i>	121
<i>SAS WHERE Clause Conditions Not Acceptable to CA-DATACOM/DB</i>	121
<i>Deciding How to Specify Selection Criteria</i>	121
<i>WHERE Clause in the View Descriptor</i>	121
<i>SAS WHERE Clause</i>	122
<i>Validation of Data Values</i>	122
<i>Validation against CA-DATADictionary</i>	122

Introduction

This appendix contains more details about some advanced topics, such as data set options, processing inserts and deletions, multiple views, and multiple windows, as well as how you can optimize your selection criteria. The discussions supplement other portions of this manual.

Data Set Options

Data set options allow you to override some of the run-time options that are stored in view descriptors. There are no LIBNAME options.

Here is a list of available data set options.

DDBKEY= 'default-key'

DDBLOAD=0|1

DDBLOCK=0|1

DDBPW= 'password'

```

DDBSV= 'status/version'
DDBTRACE=0|1
DDBURT= 'User-Requirements-Table-name'
DDBUSER= 'userid'

```

You can specify the data set options with the DATA= argument in any PROC statement or DATA step. They are effective for that single execution of the procedure. Data set options will override the corresponding values stored in the view descriptor.

The following example executes the FSEDIT procedure using a view descriptor named CUSTM. The data set options specified in the PROC statement will set the userid to JOHN and the password to MINE, regardless of what userid and password may have been included in the view descriptor.

```

proc fsedit data=vlib.custm (ddbuser='john' ddbpw='mine');
run;

```

A description of each data set option follows.

DDBKEY= Data Set Option

```
DDBKEY='default-key'
```

The DDBKEY= data set option allows you to override the Default Key specified in the view descriptor. You must specify a CA-DATACOM/DB short name for the key that you want to use.

DDBLOAD= Data Set Option

```
DDBLOAD=0|1
```

The DDBLOAD= data set option allows you to specify a fast-loading process. DDBLOAD=0 is the default, which causes the software to reposition when adding new records. Setting DDBLOAD to a non-zero value, for example, DDBLOAD=1, avoids the processing time needed to reposition on records when it is not necessary. For example, executing the APPEND procedure with the DDBLOAD = 1 would decrease processing time.

Do not set DDBLOAD equal to 1 for SAS procedures such as FSEDIT that reposition on a newly added data record.

DDBLOCK= Data Set Option

```
DDBLOCK=0|1
```

The DDBLOCK= data set option is another way to give a procedure static data. DDBLOCK=0 (the default) causes the interface to create an intermediate file (spool file), which holds the data to be processed by the SAS program. If DDBLOCK is non-zero, for example, DDBLOCK=1, the interface view engine does not build a spool file; it assumes that only one user is accessing the table or that the table is locked by some method separate from the SAS System. For more information about member-level locking and spool files, see “Locks and the Spool File” on page 112.

DDBPW= Data Set Option

```
DDBPW='password'
```

The DDBPW= data set option specifies an optional CA-DATADictionary password. If one is specified in the view descriptor, this data set option overrides it. If CA-DATADictionary requires a password and the view descriptor does not include one, you must specify the password with the DDBPW= data set option.

Not every userid requires a password. The value is the 12-character PASSWORD attribute of the PERSON entity-occurrence for the specified userid.

DDBSV= Data Set Option

DDBSV= *'status/version'*

The DDBSV= data set option specifies the status and version of the CA-DATACOM/DB table you want to access. Status/version is a 4-character field. A status is either PROD, TEST, or T (for TEST) plus a 3-digit number. The default is PROD. Other status values, such as HIST, are not allowed. Status/version can be changed any time.

DDBTRACE= Data Set Option

DDBTRACE=0 | 1

The DDBTRACE= data set option is used to analyze problems in SAS software. It also allows you to display the WHERE clause that is passed to CA-DATACOM/DB, and a message indicating whether a temporary index will be created. The default for DDBTRACE is 0, which means no information is displayed. (For more information on the WHERE clause display, see “Error Messages and Debug Information” on page 114.)

The DDBTRACE= data set option can be used to produce traces for debugging purposes. Contact Technical Support at the Institute if you need more information.

DDBURT= Data Set Option

DDBURT= *'User-Requirements-Table-name'*

The DDBURT= data set option specifies the URT that will be used by the interface view engine when it opens the view descriptor. The interface view engine will create a default URT unless you stored one as a system option, included one in the view descriptor, or specified one with the DDBURT= data set option in the SAS program.

DDBUSER= Data Set Option

DDBUSER= *'userid'*

The DDBUSER= data set option specifies the CA-DATADictionary userid. The userid is required. If you specify this data set option, it overrides the userid in the view descriptor.

The userid is the 32-character entity-occurrence name of a PERSON entity in the CA-DATADictionary database. It is not necessarily the same as the user's TSO id. The userid and optional password must have retrieval authority on six entity-types: DATABASE, FILE, RECORD, KEY, ELEMENT, and FIELD.

Using Multiple View Descriptors

The SAS/ACCESS interface to CA-DATACOM/DB supports having multiple views (view descriptors) open simultaneously. Multiple views can be open for updating as well as retrievals.

User Exits

Two user exits are available to you. You can set your own default SAS names with a user exit in the ACCESS procedure. You can also do security checking with a user exit that is available in the interface view engine. For more information on user exits, contact your SAS Technical Support Representative.

Deletions and Insertions

Deleting and inserting data records in a CA-DATACOM/DB table is a straightforward process for the SAS System as well as for CA-DATACOM/DB. However, several considerations are worth noting.

- When inserting a new row in the SAS System, you normally visualize the new data being appended at the end of the SAS data file. However, CA-DATACOM/DB conventions affect the appearance of a procedure such as FSEDIT.

When you include a WHERE clause or criteria that require a Select file, the customary SAS behavior applies. Otherwise, newly added data records appear in order by the Native Key, not at the end. The new records are always inserted in the table in the proper place, in order by the Native Key. Including or not including a WHERE clause simply determines how the table looks to the SAS System.

Note: If you are going to use a view descriptor to insert new data records, you must select at least one key (or the fields that make up the key) for that view descriptor. Select the Master Key if the Master Key values must be unique. If you request repositioning (that is, the DDBLOAD= option is set to zero or allowed to default to zero), either the default key must be selected or all of the fields that it comprises must be selected. An ADD cannot be processed if DDBLOAD equals 0 and the Default Key field is set to blanks. \triangle

- If INCLUDE-NIL-KEY equals YES, null values are indexed. However, if DUPE-MASTER-KEY equals NO, duplicate values cannot be indexed. That is, if you insert two records with missing values for the Master Key, the second insert fails. Native Key values must be unique, another reason that the second insert of a missing value fails.
- In order for you to even see newly added records while in an interactive procedure, the interface view engine does some fairly complex processing that is referred to as repositioning. Repositioning means retrieving a record immediately after it has been inserted, so that it is visible to you. This is important in an interactive procedure, but not as important in a procedure such as PROC APPEND. Therefore, to save processing time, a DDBLOAD= data set option is available that allows you to turn repositioning on and off. Setting the DDBLOAD= data set option causes the APPEND procedure, for example, to process faster and more efficiently. For more information on the DDBLOAD= data set option, see “Data Set Options” on page 117.

- Deleting a data record removes the record from the CA-DATACOM/DB table. The deleted data record is no longer available for access even though it may have satisfied the WHERE clause selection criteria before it was removed.

Missing Values (Nils)

When the interface view engine stores a missing value in a CA-DATACOM/DB table, the missing value is represented by all blanks for both numeric and character type fields. When the engine retrieves a missing value, it appears as all blanks for SAS character columns and as a period (.) for SAS numeric columns.

Note: A system option is available if you want to set missing values to zeros. See “System Options” on page 115. Δ

SAS WHERE Clause Conditions Not Acceptable to CA-DATACOM/DB

Here is a list of some (but not all) SAS WHERE clause conditions that are not acceptable to CA-DATACOM/DB; they are handled automatically by SAS post-processing.

- arithmetic expressions, for example,
 - where c1=c4*3
 - where c4<-c5
- expressions in which a column or combination of columns assumes a value of 1 or 0 to signify true or false, for example,
 - where c1
 - where (c1=c2)*20
- concatenation of character columns
- truncated comparison, for example,
 - c1=:abc
- DATETIME and TIME formats, for example,
 - '12:00't
 - '01jan60:12:00'dt
- SOUNDIX
- HAVING, GROUP BY
- references to missing values. This includes the period (.) for numeric columns, blanks for character columns, and the IS MISSING and IS NULL operators.

Deciding How to Specify Selection Criteria

Use the following guidelines to determine when to use a WHERE clause in the view descriptor and when to use a SAS WHERE clause.

WHERE Clause in the View Descriptor

Include a WHERE clause in your view descriptor when you want to

- restrict users of view descriptors to certain subsets of data
- use CA-DATACOM/DB syntax and functionality, such as masking or delimiters
- prevent users from sequentially passing the entire CA-DATACOM/DB table.

SAS WHERE Clause

Use a SAS WHERE clause when the previous guidelines do not apply and you want to

- have more run-time flexibility in subsetting data
- use SAS WHERE clause capabilities that CA-DATACOM/DB does not support, such as arithmetic expressions or truncated comparisons.

Note: Masking and delimiter characters are ignored in a SAS WHERE clause. For example,

```
where x='\y\a'
```

would cause the software to look for backslashes in the data. Δ

Validation of Data Values

The interface view engine does not validate the incoming data. That is, it does not check to see whether invalid data were entered in a CA-DATACOM/DB table, for example, nonpacked data into a packed type field. However, it does check for SAS format return codes from retrievals and moves missing values into the retrieved row if errors occur. No checking takes place for updates or WHERE clause processing.

Validation against CA-DATADictionary

The interface view engine always reads in the CA-DATADictionary information when it processes a view descriptor, because CA-DATADictionary contains information that is not in the view descriptor, for example, the element entity.

Consider the following matters regarding CA-DATADictionary usage by the interface view engine:

- Elements are groupings of fields. They are used to read and update the tables. The interface view engine makes a "best-fit" list of elements to use when it executes the view descriptor. If the Database Administrator fine-tunes the element definitions to improve performance, the interface view engine can take advantage of the improvements.
- The interface view engine rejects a view descriptor if certain information that is stored in the view descriptor does not match the dictionary. Specifically, you cannot change the following information:
 - entity-occurrence names and types for these entities: DATABASE, FILE, RECORD, KEY, FIELD
 - field class, from simple to compound or vice versa
 - field data types, from one category to another. The categories are
 - numeric:* types B (binary), 2 (halfword), 4 (fullword), 8 (double), N(Zoned), D (packed decimal)

float: types L (long float), S (short float)

alpha: types C (char), H (hex)

Xtype: types B > 8 bytes, D and N > 16 bytes, E (extended float), G (graphics), K (Kanji), T (PL/1 bit), Y (double byte character set), Z (mixed DBCS and single byte)

Note: Xtype is an artificial type. It represents a CA-DATACOM/DB type that cannot be fully handled in the SAS System. △

- You should set up a CA-DATADictionary userid and password for the SAS System and not change them. The userid should have retrieval authority in CA-DATADictionary for six entity-types: DATABASE, FILE, RECORD, ELEMENT, KEY, and FIELD. The userid and password do not protect data, but the interface view engine needs them to obtain information from CA-DATADictionary. If you change the userid or password, then the view descriptors must be recreated, or you must use the data set options to override the userid and password. (You can also zap these values in the CSECT DDBAUSE, which is used by the ACCESS procedure.)

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ACCESS Interface to CA-DATACOM/DB Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 170.

SAS/ACCESS Interface to CA-DATACOM/DB Software: Reference, Version 8

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

ISBN 1-58025-545-0

All rights reserved. Printed 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 by the government is subject to restrictions as 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, October 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.

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.