



CHAPTER

2

CA-DATACOM/DB Essentials

<i>Introduction</i>	7
<i>CA-DATACOM/DB and CA-DATADictionary Software</i>	7
<i>CA-DATACOM/DB Databases</i>	8
<i>Database Names</i>	9
<i>Tables</i>	9
<i>Fields, Keys, and Elements</i>	10
<i>Data Types</i>	10
<i>Numeric Data Types</i>	11
<i>Character Data Types</i>	11
<i>Nils-Missing Values</i>	11
<i>CA-DATACOM/DB Indexing</i>	12
<i>Selecting a Subset of Data</i>	12
<i>Sorting Data</i>	12
<i>Security Features</i>	13
<i>CA-DATACOM/DB Execution Environments</i>	13

Introduction

This chapter introduces SAS System users to CA-DATACOM/DB. It focuses on the terms and concepts that will help you use the SAS/ACCESS interface to CA-DATACOM/DB. It includes descriptions of the following:

- CA-DATACOM/DB and CA-DATADictionary software
- CA-DATACOM/DB databases, tables, records, and fields
- CA-DATACOM/DB data types
- CA-DATACOM/DB security features
- CA-DATACOM/DB indexing.

If you want more information than this chapter provides on a CA-DATACOM/DB concept or term, see the appropriate CA-DATACOM/DB documentation. For more information on CA-DATACOM/DB considerations, see Appendix 1, “Information for the Database Administrator,” on page 105 and Appendix 2, “Advanced User Topics,” on page 117.

CA-DATACOM/DB and CA-DATADictionary Software

CA-DATACOM/DB is a *database management system* (DBMS). Databases are fully defined with CA-DATADictionary.

A CA-DATACOM/DB database consists of various entity-types, which can occur one or more times. For example, a database has areas, files, and records. Each data record in a table has one or more fields. The order of the data records is determined by the values for the field specified as the Native Key. Each field contains one type of data, and each record can hold one data value for each field, except that a repeating field can assume many values.

Note: CA-DATACOM/DB data views are not supported by the SAS/ACCESS interface to CA-DATACOM/DB. △

CA-DATADITIONARY is a central, integrated, and active control facility, which provides the basis for shared and consistent system resource management. As a repository for descriptive data, CA-DATADITIONARY is your tool for

- managing definitions and syntax
- enforcing naming conventions
- creating data relationships
- managing test and production environments.

CA-DATADITIONARY allows you to collect information in categories called entity-types. Any data you enter into CA-DATADITIONARY are associated with a category, that is, an entity. For example, DATABASE, AREA, and FIELD are some specific CA-DATADITIONARY entities.

Each instance within the entity is an entity-occurrence. For example, defining a database involves storing information about the database in the DATABASE occurrence. Each database is listed by its unique name as an occurrence of the DATABASE entity.

Each occurrence has specific attributes such as data type. These attributes enable you to describe specific properties of each occurrence. For example, you can specify the type of data that a field contains or whether a key is a Master Key or a Native Key. The actual information you store for each attribute is an attribute value. In addition, CA-DATADITIONARY allows you to define support data, such as aliases, CA-DATACOM/DB descriptors, text, and relationships.

CA-DATADITIONARY allows you to have many copies (versions) of the same occurrence. Each version of an occurrence can have one status at a given time. Only TEST and PROD are allowed with the SAS/ACCESS interface to CA-DATACOM/DB. The five status values are

- TEST
- PRODUCTION
- HISTory
- INCOmplete
- QUALified production.

Depending on your site, you may find that using CA-DATADITIONARY online is more efficient for some tasks, while other tasks are simpler with batch jobs.

For more information about CA-DATADITIONARY and other CA-DATACOM/DB features or administration of CA-DATACOM/DB databases, see the appropriate CA-DATACOM/DB documentation.

CA-DATACOM/DB Databases

A CA-DATACOM/DB database is a collection of CA-DATACOM/DB tables, organized within certain CA-DATACOM/DB areas and files. Each table consists of records that have one or more FIELD entity-occurrences.

You need to know about several types of CA-DATACOM/DB entity-types to use the SAS/ACCESS interface to CA-DATACOM/DB. The most important entity-types are databases and records, which contain fields. Fields contain the actual data values, which are either character or numeric type.

You can define a field as a simple field or a compound field. Fields can also become keys, and two special keys, the Native Key and the Master Key, are required for each table. CA-DATACOM/DB generates an index for each key field. Knowing about the Native Key and the indexes can help you minimize CA-DATACOM/DB processing time for your view descriptors. In addition, fields can repeat. For more information about fields, see “Fields, Keys, and Elements” on page 10.

Typically, a database is organized according to the types of data and how you want to use the data. You must understand and be familiar with your database’s organization in order to retrieve and update information accurately and efficiently. And you must be familiar with the organization and contents of the database to create descriptor files for the SAS/ACCESS interface.

The following sections describe the various CA-DATACOM/DB entity-types that pertain to the SAS/ACCESS interface to CA-DATACOM/DB.

Database Names

Each DATABASE entity-occurrence in the CA-DATADictionary database has a unique name, from 1 to 32 characters long. A database also has a status (TEST or PROD) and version associated with it.

Tables

A *table* consists of some number of records, each having one or more fields. The table name is the name of a RECORD entity-occurrence, up to 32 characters long. Data records in the table are ordered by the values for an assigned field called the Native Key. CA-DATACOM/DB permits up to 240 tables in a database. The tables can be spread across one or more CA-DATACOM/DB areas. When you define a record for a table, you must define at least one field, one key, and one element for that record.

To create descriptor files for the SAS/ACCESS interface, you must know the name of the RECORD entity-occurrence (table) and the userid and optional password for CA-DATADictionary. An access descriptor and its associated view descriptors pertain to only one table.

Output 2.1 on page 10 illustrates four fields from the table CUSTOMERS. Field names are shown at the top of the columns. Each row represents the values in a record. The first field, CUSTOMER, is the Native Key in this table, which causes the records to be maintained in order by customer number.

Output 2.1 A Sample CA-DATACOM/DB Table

CUSTOMER	CITY	STATE	COUNTRY
14324742	San Jose	CA	USA
14569877	Memphis	TN	USA
14898029	Rockville	MD	USA
24589689	Belgrade		Yugoslavia
26422096	La Rochelle		France
38763919	Buenos Aires		Argentina
46783280	Singapore		Singapore

Fields, Keys, and Elements

Each FIELD entity-occurrence has a name (of up to 32 characters) and specific attributes, such as the data type. For more information on data types, see “Data Types” on page 10.

You can define several kinds of fields, as described briefly here.

- A *simple field* is a single field.
- A *compound field* consists of two or more simple or compound fields. The fields can be of different data types and lengths; they can also repeat or be within repeating fields. The fields making up a compound field must be contiguous.
- A *key field* allows you to quickly select and sequence data records. A key field can be any combination of simple and compound fields, up to 180 characters. The fields in a key do not have to be contiguous.
- The *Native Key* is the field that determines the order of the records in a CA-DATACOM/DB table. Each table must have one Native Key. It can be the same as the Master Key.
- The *Master Key* allows you to prevent duplicate values in a key field and to prevent changing values in that key. Each record must have one Master Key. It can be the same as the Native Key.
- A *repeating field* is a simple field or a compound field that can occur more than once. Repeating fields can also be nested within other repeating fields.
- An *element* is a unit of transfer between application programs and CA-DATACOM/DB. It consists of one or more contiguous fields. An element should contain only those fields that an application program uses at execution time. When defining an element, group together fields that are frequently accessed together in applications.

Data Types

A CA-DATACOM/DB field can be any one of a variety of data types; they are all basically type character or type numeric, as discussed below.

When you create a view descriptor, the ACCESS procedure assigns SAS formats, informats, and so on, in addition to SAS column names from the CA-DATACOM/DB

field names. See “ACCESS Procedure Data Conversions” on page 99 for the default SAS column formats and informats for each CA-DATACOM/DB data type. You can change the default formats and informats.

Numeric Data Types

Here are some of the numeric types available for CA-DATACOM/DB fields:

B	Binary
D	Packed decimal
E	Extended floating-point
L	Long floating-point
N	Numeric (zoned decimal)
S	Short floating-point
2	Halfword binary (aligned)
4	Fullword binary (aligned)
8	Doubleword binary (aligned)

Character Data Types

Here are some of the character types available for CA-DATACOM/DB fields:

C	Character
G	Graphics data
H	Hex character
K	Kanji (same as type Y)
T	PL/I bit representation
Y	Double byte character set (DBCS)
Z	Mixed DBCS and single byte

Note: CA-DATACOM/DB does not support date types. If you store dates in a CA-DATACOM/DB numeric field, identify the representation with a SAS date format in the DB Content field in your descriptor files. △

Nils–Missing Values

Missing values in a CA-DATACOM/DB table are referred to as *nil values* or simply *nils*. Nil values for both character and numeric type data are blanks, that is, HEX (40)s. All fields of a key must contain blanks for a value to be nil. There are no valid packed decimal or zoned decimal nil values. Optionally, you can specify binary zeros for nils (see “System Options” on page 115).

In the SAS System, nils are referred to as *missing values*. CA-DATACOM/DB and the SAS System handle missing values differently, but the interface view engine takes care of the differences. See “Missing Values (Nils)” on page 121 for a discussion of the differences.

CA-DATACOM/DB Indexing

An Index Area is required for each CA-DATACOM/DB database. CA-DATACOM/DB creates an index entry for each key value in each record. The indexes allow you to retrieve records quickly based on the record's contents.

Selecting a Subset of Data

A database would not be very efficient if all records had to be accessed when you needed data from only some of them. Therefore, you can specify selection criteria to identify those parts of the CA-DATACOM/DB table that you want to access.

Selection criteria contain one or more conditions that values must meet. Typically, a condition consists of a field name, an operator, and a value, but you can also compare the values of two fields or give a range of values. Conditions can be combined with AND (&) or OR (|).

Here are some sample conditions.

```
cost<.50
lastname eq 'Smith'

part=9567 & onhand>2.0e+6
```

For the SAS/ACCESS interface to CA-DATACOM/DB, you can include a WHERE clause in a view descriptor to specify selection criteria or you can include a SAS WHERE clause in a SAS program. Or you can include both WHERE clauses. The interface view engine translates WHERE clauses into CA-DATACOM/DB selection criteria.

Note that the WHERE clause for a view descriptor and the SAS WHERE clause have some differences. For more information on WHERE clauses and a description of the syntax, see "WHERE Clause in a View Descriptor" on page 91 and "Deciding How to Specify Selection Criteria" on page 121.

Sorting Data

Records in a CA-DATACOM/DB table are maintained in order by values in the specified Native Key. In a SAS/ACCESS view descriptor, you can provide a different Default Key for the view, and the records will then assume the order of your specified Default Key. You can also specify a *SORT clause*, which consists of the keyword SORT followed by one or more field names, separated by commas. You can specify ascending or descending order for each sort key; the default is ascending order. For example,

```
sort state, city, lastname desc
```

In addition, you can specify data order in a SAS program using a SAS BY clause. Note, however, that a SAS BY clause overrides a SORT clause stored in a view descriptor unless the SAS procedure includes the NOTSORTED option. In this situation, the SAS BY clause is ignored, and the SORT clause in the view descriptor is used.

For more information on SORT clauses, see "SORT Clause in a View Descriptor" on page 97.

Security Features

The CA-DATADictionary database is protected by userids, passwords, and locks. You must give the correct userid and optional password to the SAS ACCESS procedure so the procedure can obtain CA-DATADictionary information for creating an access descriptor. CA-DATACOM/DB also has security interfaces to packages such as RACF. In addition, you can develop your own security program through a user exit in the interface view engine.

CA-DATACOM/DB Execution Environments

When you access a CA-DATACOM/DB database, you can work in either a Single User execution environment or a Multi-User execution environment. In a *Single User environment*, each user has a copy of CA-DATACOM/DB and has exclusive use of the database.

The SAS/ACCESS interface to CA-DATACOM/DB runs only under a *Multi-User environment*. In a Multi-User environment, many databases can be accessed from many regions concurrently, with exclusive control at the logical record level. Only one copy of CA-DATACOM/DB is required to handle all regions. Recovery is centralized for all users.

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.