



## APPENDIX

## 3

## Data and Descriptors for the Examples

---

<i>Introduction</i>	125
<i>CA-DATACOM/DB Tables</i>	126
<i>CA-DATADictionary Statements for Sample Tables</i>	126
<i>CUSTOMERS Table</i>	132
<i>EMPLOYEES Table</i>	135
<i>INVOICE Table</i>	136
<i>ORDER Table</i>	138
<i>Access Descriptors for the CA-DATACOM/DB Tables</i>	140
<i>MYLIB.CUSTS Access Descriptor</i>	140
<i>MYLIB.EMPLOYEE Access Descriptor</i>	141
<i>MYLIB.INVOICE Access Descriptor</i>	141
<i>MYLIB.ORDERES Access Descriptor</i>	142
<i>View Descriptors for the CA-DATACOM/DB Tables</i>	142
<i>VLIB.ALLEMP View Descriptor</i>	142
<i>VLIB.ALLODR View Descriptor</i>	142
<i>VLIB.CUSORDR View Descriptor</i>	142
<i>VLIB.CUSPHON View Descriptor</i>	143
<i>VLIB.CUSTADD View Descriptor</i>	143
<i>VLIB.DCMEMPS View Descriptor</i>	143
<i>VLIB.EMPINFO View Descriptor</i>	143
<i>VLIB.EMPS View Descriptor</i>	143
<i>VLIB.FORINV View Descriptor</i>	144
<i>VLIB.INV View Descriptor</i>	144
<i>VLIB.USACUST View Descriptor</i>	144
<i>VLIB.USAINV View Descriptor</i>	144
<i>VLIB.USAORDR View Descriptor</i>	145
<i>SAS Data Files</i>	145
<i>MYDATA.OUTOFSTK Data File</i>	145
<i>MYDATA.SASEMPS Data File</i>	146
<i>LIB6.BIRTHDAY Data File</i>	147

---

## Introduction

This appendix gives information about the CA-DATACOM/DB tables, descriptor files, and SAS data files used in the examples in this book. It shows the CA-DATADictionary statements and the data that were used to build the CA-DATACOM/DB tables. It also displays the access descriptors and view descriptors, along with selection criteria specified for them. In addition, this appendix shows the data and the SAS statements that were used to create the SAS data files for the examples.

If you want to run the examples, contact your SAS Software Representative for information.

---

## CA-DATACOM/DB Tables

This section describes the CA-DATACOM/DB tables referenced in this book. It shows

- the CA-DATADictionary statements and data for the tables
- the Native Key and the Master Key for each table
- the compound fields and repeating fields, if any.

The four CA-DATACOM/DB tables used in the examples are named CUSTOMERS, EMPLOYEES, INVOICE, and ORDER. They are all in one CA-DATACOM/DB database named TEXTILES. To build these tables, follow the steps shown below.

- 1 Create the CA-DATADictionary entries and catalog them to CA-DATACOM/DB.
- 2 Create the SAS data files.
- 3 Create an access descriptor and an associated view descriptor for each table. Make sure that all SAS names in the view descriptors match the names in the SAS data files. Use the access descriptors in this appendix as a model. Select every field for the access descriptors, and create view descriptors that also select every field.
- 4 Run the APPEND procedure with the data set options shown here:

```
proc append data=SAS-file base=view-descriptor; run;
```

The CA-DATADictionary statements and the data for the tables are shown next.

---

## CA-DATADictionary Statements for Sample Tables

Here are the CA-DATADictionary statements used to create the four sample CA-DATACOM/DB tables. This is input to the DDUPDATE utility.

```
//SYSIN DD *
-USR ADR-INSTALL,NEWUSER
-ADD DATABASE,TEXTILES(T001)
3000 030
-END
-ADD AREA,TEX030(T001)
1000 CONNECT,TEXTILES
3001 TEX SASBxB.DATACOM.TEX030
3002     TEX030           3380   04096
-END
-ADD FILE,CUSTOMF(T001)
1000 CONNECT,TEX030
3100 DB           01024           FBLK Y
3101 CUS 001 Y Y           ADR/DB
-END
-ADD RECORD,CUSTOMERS(T001)
1000 CONNECT,CUSTOMF
3200 DB
-END
-GRP START,RECORD,CUSTOMERS(T001)
-ADD FIELD,CUSTOMER
4010 START                               START
```

```

4012 S C L N 00008 00 00001
-ADD FIELD,STATEZIP
4010 CUSTOMER START
4012 C C L N 00007 00 00001
-ADD FIELD,STATE
4010 STATEZIP STATEZIP
4012 S C L N 00002 00 00001
-ADD FIELD,ZIPCODE
4010 STATE STATEZIP
4012 S N R N 00005 00 00001
-ADD FIELD,COUNTRY
4010 STATE START
4012 S C L N 00020 00 00001
-ADD FIELD,TELEPHONE
4010 COUNTRY START
4012 S C L N 00012 00 00001
-ADD FIELD,NAME
4010 TELEPHONE START
4012 S C L N 00060 00 00001
-ADD FIELD,CONTACT
4010 NAME START
4012 S C L N 00030 00 00001
-ADD FIELD,STREETADDRESS
4010 CONTACT START
4012 S C L N 00040 00 00001
-ADD FIELD,CITY
4010 STREETADDRESS START
4012 S C L N 00025 00 00001
-ADD FIELD,FIRSTORDERDATE
4010 CITY START
4012 S N R N 00006 00 00001
-ADD FIELD,SIGNATURELIST
4010 FIRSTORDERDATE START
4012 C C L N 00044 00 00005
-ADD FIELD,LIMIT
4010 SIGNATURELIST SIGNATURELIST
4012 S N R N 00014 02 00001
-ADD FIELD,SIGNATURE
4010 LIMIT SIGNATURELIST
4012 S C L N 00030 00 00001
-ADD FIELD,SIGNATURE
4010 LIMIT SIGNATURELIST
4012 S C L N 00030 00 00001
-ADD FIELD,SIGNATURE
4010 LIMIT SIGNATURELIST

```

```

-ADD FIELD, BRANCHOFFICE
4010 SIGNATURE START
4012 S C L N 00025 00 00010
-END
-GRP END
-ADD KEY, CUSTOMERS.CUSKEY(T001)
5000 CUSKY 001 Y Y Y
5010 ADD CUSTOMER
5011 $FIRST
-END
-ADD ELEMENT, CUSTOMERS.CUSELM(T001)
6000 CUSEL
6010 ADD CUSTOMER
6010 ADD STATE
6010 ADD ZIPCODE
6010 ADD COUNTRY
6010 ADD TELEPHONE
6010 ADD NAME
6010 ADD CONTACT
6010 ADD STREETADDRESS
6010 ADD CITY
6010 ADD FIRSTORDERDATE
6010 ADD SIGNATURELIST
6010 ADD BRANCHOFFICE
-END
-ADD FILE, EMPLOYF(T001)
1000 CONNECT, TEX030
3100 DB 01024 FBLK Y
3101 EMP 002 Y Y ADR/DB
-END
-ADD RECORD, EMPLOYEES(T001)
1000 CONNECT, EMPLOYF
3200 DB
-END
-GRP START, RECORD, EMPLOYEES(T001)
-ADD FIELD, EMPID
4010 START START
4012 S N R N 00006 00 00001
-END
-ADD FIELD, HIREDATE
4010 EMPID START
4012 S N R N 00006 00 00001
-END
-ADD FIELD, SALARY
4010 HIREDATE START
4012 S N R N 00007 02 00001
-END
-ADD FIELD, DEPT
4010 SALARY START
4012 S C L N 00006 00 00001
-END
-ADD FIELD, JOBCODE
4010 DEPT START
4012 S N R N 00005 00 00001

```

```

-END
-ADD FIELD,SEX
4010 JOBCODE                                START
4012 S C L N 00001 00 00001
-END
-ADD FIELD,BIRTHDATE
4010 SEX                                    START
4012 S N R N 00006 00 00001
-END
-ADD FIELD,LASTNAME
4010 BIRTHDATE                             START
4012 S C L N 00018 00 00001
-END
-ADD FIELD,FIRSTNAME
4010 LASTNAME                              START
4012 S C L N 00015 00 00001
-END
-ADD FIELD,MIDDLENAME
4010 FIRSTNAME                             START
4012 S C L N 00015 00 00001
-END
-ADD FIELD,PHONE
4010 MIDDLENAME                            START
4012 S C L N 00004 00 00001
-END
-GRP END
-ADD KEY,EMPLOYEES.EMPKEY(T001)
5000 EMPKY 002 Y Y                          Y
5010 ADD EMPID
5011 $FIRST
-END
-ADD ELEMENT,EMPLOYEES.EMPELM(T001)
6000 EMPEL
6010 ADD EMPID
6010 ADD HIREDATE
6010 ADD SALARY
6010 ADD DEPT
6010 ADD JOBCODE
6010 ADD SEX
6010 ADD BIRTHDATE
6010 ADD LASTNAME
6010 ADD FIRSTNAME
6010 ADD MIDDLENAME
6010 ADD PHONE
-ADD FILE,INVOICF(T001)
1000 CONNECT,TEX030
3100 DB          01024          FBLK Y
3101 INV 003 Y Y          ADR/DB
-END
-ADD RECORD,INVOICE(T001)
1000 CONNECT,INVOICF
3200 DB
-END
-GRP START,RECORD,INVOICE(T001)

```

```

-ADD FIELD, INVOICENUM
4010 START START
4012 S N R N 00005 00 00001
-ADD FIELD, BILLEDTO
4010 INVOICENUM START
4012 S C L N 00008 00 00001
-ADD FIELD, AMTBILLED
4010 BILLEDTO START
4012 S N R N 00014 02 00001
-ADD FIELD, COUNTRY
4010 AMTBILLED START
4012 S C L N 00020 00 00001
-ADD FIELD, AMOUNTINUS
4010 COUNTRY START
4012 S N R N 00010 02 00001
-ADD FIELD, BILLEDBY
4010 AMOUNTINUS START
4012 S N R N 00006 00 00001
-ADD FIELD, BILLEDON
4010 BILLEDBY START
4012 S N R N 00006 00 00001
-ADD FIELD, PAIDON
4010 BILLEDON START
4012 S N R N 00006 00 00001
-ADD FIELD, COMPUTEDEXCHANG
4010 PAIDON START
4012 S L R Y 00008 00 00001
-GRP END
-ADD KEY, INVOICE.INVKEY(T001)
5000 INVKY 003 Y Y Y
5010 ADD INVOICENUM
5011 $FIRST
-ADD ELEMENT, INVOICE.INVELM(T001)
6000 INVEL
6010 ADD INVOICENUM
6010 ADD BILLEDTO
6010 ADD AMTBILLED
6010 ADD COUNTRY
6010 ADD AMOUNTINUS
6010 ADD BILLEDBY
6010 ADD BILLEDON
6010 ADD PAIDON
6010 ADD COMPUTEDEXCHANG
-ADD FILE, ORDERF(T001)

```

```

1000 CONNECT,TEX030
3100 DB          01024          FBLK Y
3101 ORD 004 Y Y          ADR/DB
-ADD RECORD,ORDER(T001)
1000 CONNECT,ORDERF
3200 DB
-ADD FIELD,ORDERNUM
4010 START
4012 S N R N 00005 00 00001
-ADD FIELD,STOCKNUM
4010 ORDERNUM
4012 S N R N 00004 00 00001
-ADD FIELD,LENGTH
4010 STOCKNUM
4012 S N R N 00004 00 00001
-ADD FIELD,FABRICCHARGES
4010 LENGTH
4012 S N R N 00010 02 00001
-ADD FIELD,SHIPTO
4010 FABRICCHARGES
4012 S C L N 00008 00 00001
-ADD FIELD,DATEORDERED
4010 SHIPTO
4012 S N R N 00006 00 00001
-ADD FIELD,SHIPPED
4010 DATEORDERED
4012 S N R N 00006 00 00001
-ADD FIELD,TAKENBY
4010 SHIPPED
4012 S N R N 00006 00 00001
-ADD FIELD,PROCESSEDBY
4010 TAKENBY
4012 S N R N 00006 00 00001
-ADD FIELD,SPECIALINSTRUCT
4010 PROCESSEDBY
4012 S C L N 00001 00 00001
-GRP END
-ADD KEY,ORDER.ORDKEY(T001)
5000 ORDKY 004 Y Y          Y
5010 ADD ORDERNUM
5011 $FIRST

```

```

-END
-ADD ELEMENT,ORDER.ORDELM(T001)
6000 ORDEL
6010 ADD ORDERNUM
6010 ADD STOCKNUM
6010 ADD LENGTH
6010 ADD FABRICCHARGES
6010 ADD SHIPTO
6010 ADD DATEORDERED
6010 ADD SHIPPED
6010 ADD TAKENBY
6010 ADD PROCESSEDBY
6010 ADD SPECIALINSTRUCT
//

```

---

## CUSTOMERS Table

The sample CA-DATACOM/DB table named CUSTOMERS is in the TEXTILES database. The userid is DEMO. CUSKEY is the Native Key and the Master Key. CUSKEY consists of the CUSTOMER field, which contains a unique integer number for each customer.

In the CUSTOMERS table, STATEZIP is a compound field, consisting of the two simple fields STATE and ZIPCODE at level 2. SIGNATURELIST is also a compound field; it consists of the LIMIT field and the SIGNATURE field. The data values for this compound field indicate whose signature is required for specific amounts of money. This compound field is also a repeating field. The last field, BRANCHOFFICE, is a repeating field that can occur ten times. It is a simple field containing city names of branch offices for the customer.

Here are the fields in the CUSTOMERS table. The data are shown in Output A3.1 on page 133.

```

DATA CUSTOMER;
      INPUT @1  CUSTNUM      $8.          /* CUSTOMER NUMBER          */
           @10 STATE        $2.
           @13 ZIPCODE      5.          /* ZIPCODE IF COMPANY IS   */
                                           /* IN THE U.S., OTHERWISE  */
                                           /* IT IS THE MAIL CODE     */
                                           /* APPROPRIATE FOR THE     */
                                           /* COUNTRY WHERE THE       */
                                           /* COMPANY IS LOCATED      */
           @20 COUNTRY      $20.
           @42 PHONE        $12. /
           @1  NAME         $60. / /* CUSTOMER'S COMPANY NAME*/
           @1  CONTACT      $30. /* CONTACT AT CUSTOMER'S  */
                                           /* COMPANY                  */
           @31 STREET       $40. /
           @1  CITY         $25.
           @30 FIRSTORD     YYMMDD6./ /* DATE OF FIRST ORDER     */
           @1  LIMIT1       15.2 /* SIGNATURE LIMIT #1      */
           @20 SIGNATU1     $30. / /* SIGNATURE NAME #1      */
           @1  LIMIT2       15.2 /* SIGNATURE LIMIT #2      */
           @20 SIGNATU2     $30. / /* SIGNATURE NAME #2      */
           @1  LIMIT3       15.2 /* SIGNATURE LIMIT #3      */
           @20 SIGNATU3     $30. / /* SIGNATURE NAME #3      */

```



```

@1 BRANCHO1 $25. /* BRANCH OFFICE #1 */
@30 BRANCHO2 $25. / /* BRANCH OFFICE #2 */
@1 BRANCHO3 $25. /* BRANCH OFFICE #3 */
@30 BRANCHO4 $25.; /* BRANCH OFFICE #4 */
FORMAT FIRSTORD DATE7.;

```

## Output A3.1 Data for CUSTOMERS Table

CUSTOMER	NAME	COUNTRY	TELEPHON
12345678			919/489-5682
14324742	SANTA CLARA VALLEY TECHNOLOGY SPECIALISTS	USA	408/629-0589
14569877	PRECISION PRODUCTS	USA	919/489-6792
14898029	UNIVERSITY BIOMEDICAL MATERIALS	USA	301/760-2541
15432147	GREAT LAKES LABORATORY EQUIPMENT MANUFACTURERS	USA	616/582-3906
18543489	LONE STAR STATE RESEARCH SUPPLIERS	USA	512/478-0788
19783482	TWENTY-FIRST CENTURY MATERIALS	USA	703/714-2900
19876078	SAN JOAQUIN SCIENTIFIC AND INDUSTRIAL SUPPLY, INC.	USA	209/686-3953
24589689	CENTAR ZA TECHNIKU I NAUCNU RESTAURIRANJE UMJETNINA	Yugoslavia	(012)736-202
26422096	SOCIETE DE RECHERCHES POUR DE CHIRURGIE ORTHOPEDIQUE	France	4268-54-72
26984578	INSTITUT FUR TEXTIL-FORSCHUNGS	Austria	43-57-04
27654351	INSTITUT DE RECHERCHE SCIENTIFIQUE MEDICALE	Belgium	02/215-37-32
28710427	ANTONIE VAN LEEUWENHOEK VERENIGING VOOR MICROBIOLOGIE	Netherlands	(021)570517
29834248	BRITISH MEDICAL RESEARCH AND SURGICAL SUPPLY	Britain	(0552)715311
31548901	NATIONAL COUNCIL FOR MATERIALS RESEARCH	Canada	406/422-3413
38763919	INSTITUTO DE BIOLOGIA Y MEDICINA NUCLEAR	Argentina	244-6324
39045213	LABORATORIO DE PESQUISAS VETERINARIAS DESIDERIO FINAMOR	Brazil	012/302-1021
43290587	HASSEI SAIBO GAKKAI	Japan	(02)933-3212
43459747	RESEARCH OUTFITTERS	Australia	03/734-5111
46543295	WESTERN TECHNOLOGICAL SUPPLY	Japan	(03)022-2332
46783280	NGEE TECHNOLOGICAL INSTITUTE	Singapore	3762855
48345514	GULF SCIENTIFIC SUPPLIES	United Arab Emirates	213445

  

CUSTOMER	STREETAD	CITY	STATE	ZIPCODE
12345678			NC	.
14324742	5089 CALERO AVENUE	SAN JOSE	CA	95123
14569877	198 FAYETTVILLE ROAD	MEMPHIS	NC	27514
14898029	1598 PICCARD DRIVE	ROCKVILLE	MD	20850
15432147	103 HARRIET STREET	KALAMAZOO	MI	49001
18543489	5609 RIO GRANDE	AUSTIN	TX	78701
19783482	4613 MICHAEL FARADAY DRIVE	RESTON	VA	22090
19876078	1095 HIGHWAY 99 SOUTH	TULARE	CA	93274
24589689	TAKOVSKA 4	BELGRADE		.
26422096	40 RUE PERIGNON	LA ROCHELLE		75014
26984578	MECHITARISTENGASSE 5	VIENNA		5110
27654351	103 RUE D'EGMONT	BRUSSELS		5010
28710427	BIRMOERSTRAAT 34	THE HAGUE	HV	3607
29834248	44 PRINCESS GATE, HYDE PARK	LONDON, SW7 1PU		.
31548901	5063 RICHMOND MALL	VANCOUVER, V5T 1L2	BC	.
38763919	SALGUERO 2345	BUENOS AIRES		1405
39045213	RUA DONA ANTONIA DE QUEIROS 381	SAO PAULO	SP	1051
43290587	3-2-7 ETCHUJMA, KOTO-KU	TOKYO 101		.
43459747	191 LOWER PLENTY ROAD	PRAHRAN, VICTORIA		3181
46543295	4-3-8 ETCHUJMA, KOTO-KU	TOKYO 102		.
46783280	356 CLEMENTI ROAD	SINGAPORE		2374
48345514	POB 8032	RAS AL KHAIMAH		.

CUSTOMER	CONTACT	FIRSTORD	LIMIT	SIGNATU
12345678		.	.	
14324742	A. BAUM	05FEB65	5000.00	BOB HENSON
			25000.00	KAREN DRESSER
14569877	CHARLES BARON	15AUG83	5000.00	JEAN CRANDALL
			100000.00	STEVE BLUNTSEN
14898029	S. TURNER	12NOV76	10000.00	MASON FOXWORTH
			50000.00	DANIEL STEVENS
			100000.00	ELIZABETH PATTON
15432147	D.W. KADARAUCH	28APR86	10000.00	JACK TREVANE
18543489	A. SILVERIA	10SEP79	10000.00	NANCY WALSH
			50000.00	TED WHISTLER
			100000.00	EVAN MASSEY
19783482	M.R. HEFFERNAN	18JUL68	5000.00	PETER THOMAS
			10000.00	LOUIS PICKERING
19876078	J.A. WHITTEN	11MAY79	7500.00	EDWARD LOWE
			25000.00	E.F. JENSEN
24589689	J.V. VUKASINOVIC	30NOV81	.	
26422096	Y. CHAVANON	14JUN83	5000.00	MICHELE PICARD
			10000.00	M.L.SEIS
26984578	GUNTER SPIELMANN	25MAY87	100000.00	FRANZ BECH
27654351	I. CLEMENS	14OCT86	5000.00	C.J. HELMER
28710427	M.C. BORGSTEEDE	10OCT85	10000.00	J.J. JASPER
29834248	A.D.M. BRYCESON	29JAN86	5000.00	ELVIN POMEROY
31548901	W.E. MACDONALD	19MAR84	1000.00	DAPHNE MARSHALL
38763919	JORGE RUNNAZZO	10DEC84	2500.00	M.L. CARLOS
39045213	ELISABETE REGIS GUILLAUMON	18AUG82	1500.00	RICK ESTABAN
43290587	Y. FUKUDA	08FEB74	10000.00	R. YAMOTO
43459747	R.G. HUGHES	28JUL72	1000.00	DENNIS RICHMOND
			5000.00	JANICE HEATH
46543295		19APR84	10000.00	DAPHNE MARSHALL
46783280	LING TAO SOON	27SEP79	.	
48345514	J.Q. RIFAI	10SEP86	.	

CUSTOMER	BRANCHO1	BRANCHO2	BRANCHO3	BRANCHO4
12345678				
14324742	TORONTO	HOUSTON	TOKYO	LONDON
14569877	NEW YORK	CHICAGO	LOS ANGELES	
14898029	NEW YORK	CHICAGO	DALLAS	
15432147	CHICAGO	COLUMBUS		
18543489	HOUSTON	DALLAS	EL PASO	LUBBOCK
19783482	WASHINGTON D.C.	NEW YORK		
19876078				
24589689				
26422096	LONDON	NEW YORK		
26984578	LONDON	NEW YORK	ROME	
27654351	LONDON	BOSTON		
28710427	LONDON			
29834248	SINGAPORE	TORONTO	CAIRO	
31548901	SEATTLE	TORONTO		
38763919	MIAMI	NEW YORK		
39045213	MIAMI	NEW YORK		
43290587	SAN FRANCISCO			
43459747	SEATTLE			
46543295	SEATTLE	TORONTO	SAN FRANCISCO	DENVER
46783280				
48345514				

---

## EMPLOYEES Table

The sample CA-DATACOM/DB table named EMPLOYEES is in the TEXTILES database. The userid is DEMO. EMPKEY is the Native Key and the Master Key. It consists of the numeric field EMPID, which contains a unique integer number for each employee.

Here are the fields in the EMPLOYEE table. The data are shown in Output A3.2 on page 136.

```
DATA EMPLOY;
  INPUT @1  EMPID      6.           /* EMPLOYEE ID NUMBER      */
        @10 HIREDATE  YYMMDD6.
        @20 SALARY    8.2
        @30 DEPT      $6.
        @40 JOBCODE   5.
        @47 SEX       $1.
        @50 BIRTHDAT  YYMMDD6. /
        @1  LASTNAME  $18.
        @20 FIRSTNAM  $15.
        @40 MIDDLENA  $15.
        @60 PHONE     $4. ;
FORMAT HIREDATE DATE7.;
FORMAT BIRTHDAT DATE7.;
```

## Output A3.2 Data for EMPLOYEES Table

OBS	EMPID	HIREDATE	SALARY	DEPT	JOBCODE	SEX	BIRTHDAT	LASTNAME
1	119012	01JUL68	42340.58	CSR010	602	F	05JAN46	WOLF-PROVENZA
2	120591	05DEC80	31000.55	SHP002	602	F	12FEB46	HAMMERSTEIN
3	123456	04APR89	.	.	.	.	.	VARGAS
4	127845	16JAN67	75320.34	ACC024	204	M	25DEC43	MEDER
5	129540	01AUG82	56123.34	SHP002	204	F	31JUL60	CHOULAI
6	135673	15JUL84	46322.58	ACC013	602	F	21MAR61	HEMESLY
7	212916	15FEB51	52345.58	CSR010	602	F	29MAY28	WACHBERGER
8	216382	15JUN85	34004.65	SHP013	602	F	24JUL63	PURINTON
9	234967	19DEC88	17000.00	CSR004	602	M	21DEC67	SMITH
10	237642	01NOV76	43200.34	SHP013	602	M	13MAR54	BATTERSBY
11	239185	07MAY81	57920.66	ACC024	602	M	28AUG59	DOS REMEDIOS
12	254896	04APR85	35000.74	CSR011	204	M	06APR49	TAYLOR-HUNYADI
13	321783	10SEP67	48931.58	CSR011	602	M	03JUN35	GONZALES
14	328140	10JAN75	75000.34	ACC043	1204	F	02JUN51	MEDINA-SIDONIA
15	346917	02MAR87	46000.33	SHP013	204	F	15MAR50	SHIEKEESLAM
16	356134	14JUN85	62450.75	ACC013	204	F	25OCT60	DUNNETT
17	423286	19DEC88	32870.66	ACC024	602	M	31OCT64	MIFUNE
18	456910	14JUN78	45000.58	CSR010	602	M	24SEP53	ARDIS
19	456921	19AUG87	33210.04	SHP002	602	M	12MAY62	KRAUSE
20	457232	15JUL85	55000.66	ACC013	602	M	15OCT63	LOVELL
21	459287	02NOV64	50000.00	SHP024	204	M	05JAN34	RODRIGUES
22	677890	12DEC88	37610.00	CSR010	204	F	24APR65	NISHIMATSU-LYNCH

  

OBS	FIRSTNAM	MIDDLENA	PHONE
1	G.	ANDREA	3467
2	S.	RACHAEL	3287
3	PAUL	JESUS	
4	VLADIMIR	JORAN	6231
5	CLARA	JANE	3921
6	STEPHANIE	J.	6329
7	MARIE-LOUISE	TERESA	8562
8	PRUDENCE	VALENTINE	3852
9	GILBERT	IRVINE	7274
10	R.	STEPHEN	8342
11	LEONARD	WESLEY	4892
12	ITO	MISHIMA	0231
13	GUILLERMO	RICARDO	3642
14	MARGARET	ROSE	5901
15	SHALA	Y.	8745
16	CHRISTINE	MARIE	4213
17	YUKIO	TOSHIRO	3278
18	RICHARD	BINGHAM	4351
19	KARL-HEINZ	G.	7452
20	WILLIAM	SINCLAIR	6321
21	JUAN	M.	5879
22	CAROL	ANNE	6245

**INVOICE Table**

The sample CA-DATACOM/DB table named INVOICE is in the TEXTILES database. The userid is DEMO. INVKEY is the Native Key and the Master Key. It consists of the numeric field INVOICE, which contains a unique integer number for each invoice.

Here are the fields for the INVOICE table. The data are shown in Output A3.3 on page 138.

```

DATA INVOICE;
  INPUT @1 INVOICEN 5.          /* INVOICE NUMBER      */
        @7 BILLEDTO $8.        /* COMPANY THAT PLACED THE
                                /* THE ORDER           */

        @15 AMTBILLE 15.2      /* AMOUNT OF BILL IN LOCAL
                                /* CURRENCY            */

        @30 COUNTRY $20.
        @50 AMOUNTIN 11.2 /    /* AMOUNT OF BILL IN U.S.
                                /* DOLLARS             */

        @1 BILLEDBY 6.         /* EMPLOYEE WHO WROTE THE
                                /* BILL                */

        @10 BILLEDON YYMMDD6.  /* DATE THAT BILL WAS SENT
                                /*                      */

        @20 PAIDON YYMMDD6.    /* DATE THAT BILL WAS PAID
                                /*                      */

        @30 COMPUTED TIME8. ;  /* TIME OF DAY THAT THE
                                /* EXCHANGE RATE TO U.S.
                                /* DOLLARS WAS COMPUTED
                                /*

FORMAT BILLEDON DATE7.;
FORMAT PAIDON DATE7.;

```

## Output A3.3 Data for INVOICE Table

OBS	INVOICEN	BILLEDTO	AMTBILLE	COUNTRY	AMOUNTIN
1	11270	39045213	1340738760.90	Brazil	2256870.00
2	11271	18543489	11063836.00	USA	11063836.00
3	11273	19783482	252148.50	USA	252148.50
4	11276	14324742	1934460.00	USA	1934460.00
5	11278	14898029	1400825.00	USA	1400825.00
6	11280	39045213	1340738760.90	Brazil	2256870.00
7	11282	19783482	252148.50	USA	252148.50
8	11285	38763919	34891210.20	Argentina	2256870.00
9	11286	43459747	12679156.00	Australia	11063836.00
10	11287	15432147	252148.50	USA	252148.50
11	12051	39045213	1340738760.90	Brazil	2256870.00
12	12102	18543489	11063836.00	USA	11063836.00
13	12263	19783482	252148.50	USA	252148.50
14	12468	14898029	1400825.00	USA	1400825.00
15	12471	39045213	1340738760.90	Brazil	2256870.00
16	12476	38763919	34891210.20	Argentina	2256870.00
17	12478	15432147	252148.50	USA	252148.50

  

OBS	BILLEDBY	BILLEDON	PAIDON	COMPUTED
1	239185	05OCT88	18OCT88	3.9646000000000000E+04
2	457232	05OCT88	11OCT88	.
3	239185	06OCT88	11NOV88	.
4	135673	06OCT88	20OCT88	.
5	239185	06OCT88	19OCT88	.
6	423286	07OCT88	20OCT88	5.8154000000000000E+04
7	457232	07OCT88	25OCT88	.
8	239185	10OCT88	30NOV88	5.5163000000000000E+04
9	423286	10OCT88	.	3.3827000000000000E+04
10	457232	11OCT88	04NOV88	.
11	457232	02NOV88	.	3.1185000000000000E+04
12	239185	17NOV88	.	.
13	423286	05DEC88	.	.
14	135673	24DEC88	02JAN89	.
15	457232	27DEC88	.	5.0945000000000000E+04
16	135673	24DEC88	.	3.9563000000000000E+04
17	423286	24DEC88	02JAN89	.

**ORDER Table**

The sample CA-DATACOM/DB table named ORDER is in the TEXTILES database. The userid is DEMO. ORDKEY is the Native Key and the Master Key. It consists of the numeric field ORDERNUM, which contains a unique integer number for each order.

Here are the fields for the ORDER table. The data are shown in Output A3.4 on page 139.

```
DATA ORDERS;
  INPUT @1 ORDERNUM 5.          /* ORDER NUMBER          */
        @6 STOCKNUM 4.         /* STOCK NUMBER          */

        @10 LENGTH 4.          /* LENGTH OF MATERIAL ORDERED */
```

```

@15 FABRICCH 11.2      /* FABRIC CHARGES          */
@27 SHIPTO   $8.      /* CUSTOMER WHOM ORDER IS TO BE */
                   /* SHIPPED TO                */
@35 DATEORDE YYMMDD6. /* DATE OF ORDER            */
@45 SHIPPED  YYMMDD6. /* DATE THAT ORDER WAS SHIPPED */
@55 TAKENBY  6.      /* EMPLOYEE WHO TOOK THE ORDER */
@62 PROCESSE 6.      /* EMPLOYEE WHO PROCESSED THE   */
                   /* THE ORDER                  */
@69 SPECIALI $1. ;   /* THIS IS A FLAG THAT SIGNALS */
                   /* THERE ARE SPECIAL INSTRUCTIONS */
                   /* ASSOCIATED WITH THIS ORDER.  */

FORMAT DATEORDE DATE7.;
FORMAT SHIPPED DATE7.;

```

Output A3.4 Data for ORDER Table

OBS	ORDERNUM	STOCKNUM	LENGTH	FABRICCH
1	11269	9870	690	.
2	11270	1279	1750	2256870.00
3	11271	8934	110	11063836.00
4	11272	3478	1000	.
5	11273	2567	450	252148.50
6	11274	4789	1000	.
7	11275	3478	1000	.
8	11276	1279	1500	1934460.00
9	11277	8934	100	10058033.00
10	11278	2567	2500	1400825.00
11	11279	9870	650	.
12	11280	1279	1750	2256870.00
13	11281	8934	110	11063836.00
14	11282	2567	450	252148.50
15	11283	9870	690	.
16	11284	3478	1000	.
17	11285	1279	1750	2256870.00
18	11286	8934	110	11063836.00
19	11287	2567	450	252148.50
20	11288	9870	690	.
21	11969	9870	690	.
22	12051	1279	1750	2256870.00
23	12102	8934	110	11063836.00
24	12160	3478	1000	.
25	12263	2567	450	252148.50
26	12464	4789	1000	.
27	12465	3478	1000	.
28	12466	1279	1500	1934460.00
29	12467	8934	100	10058033.00
30	12468	2567	2500	1400825.00
31	12470	9870	650	.
32	12471	1279	1750	2256870.00
33	12472	8934	110	11063836.00
34	12473	2567	450	252148.50
35	12474	9870	690	.
36	12475	3478	1000	.
37	12476	1279	1750	2256870.00
38	12477	8934	110	11063836.00
39	12478	2567	450	252148.50
40	12479	9870	690	.

OBS	ORDERNUM	SHIPTO	DATEORDE	SHIPPED	TAKENBY	PROCESSE	SPECIALI
1	11269	19876078	03OCT88	.	212916	.	
2	11270	39045213	03OCT88	19OCT88	321783	237642	X
3	11271	18543489	03OCT88	13OCT88	456910	456921	
4	11272	29834248	03OCT88	.	234967	.	
5	11273	19783482	04OCT88	14NOV88	119012	216382	
6	11274	15432147	04OCT88	.	212916	.	
7	11275	29834248	04OCT88	.	234967	.	
8	11276	14324742	04OCT88	21OCT88	321783	120591	X
9	11277	31548901	05OCT88	.	456910	.	
10	11278	14898029	05OCT88	20OCT88	119012	456921	
11	11279	48345514	05OCT88	.	212916	.	
12	11280	39045213	06OCT88	21OCT88	321783	237642	X
13	11281	18543489	06OCT88	27OCT88	456910	216382	
14	11282	19783482	06OCT88	26OCT88	119012	456921	
15	11283	18543489	07OCT88	.	212916	.	
16	11284	24589689	07OCT88	.	234967	.	
17	11285	38763919	07OCT88	02DEC88	321783	120591	X
18	11286	43459747	07OCT88	03NOV88	456910	237642	
19	11287	15432147	07OCT88	07NOV88	119012	216382	
20	11288	14324742	10OCT88	.	212916	.	Y
21	11969	19876078	25OCT88	.	212916	.	
22	12051	39045213	31OCT88	.	321783	.	X
23	12102	18543489	15NOV88	.	456910	.	
24	12160	29834248	19NOV88	.	234967	.	Z
25	12263	19783482	01DEC88	.	119012	.	
26	12464	15432147	23DEC88	.	212916	.	
27	12465	29834248	23DEC88	.	234967	.	
28	12466	14324742	23DEC88	.	321783	.	X
29	12467	31548901	23DEC88	.	456910	.	
30	12468	14898029	23DEC88	03JAN89	119012	120591	
31	12470	48345514	23DEC88	.	212916	.	
32	12471	39045213	23DEC88	.	321783	.	X
33	12472	18543489	23DEC88	03JAN89	456910	237642	
34	12473	19783482	23DEC88	.	119012	.	
35	12474	18543489	23DEC88	.	212916	.	
36	12475	24589689	23DEC88	.	234967	.	
37	12476	38763919	23DEC88	03JAN89	321783	456921	X
38	12477	43459747	23DEC88	.	456910	.	
39	12478	15432147	23DEC88	03JAN89	119012	216382	
40	12479	14324742	23DEC88	.	212916	.	

---

## Access Descriptors for the CA-DATACOM/DB Tables

This section shows the access descriptors for the four CA-DATACOM/DB tables used in the examples. The tables are described in "CA-DATACOM/DB Tables" on page 126. The access descriptors are presented in alphabetical order for easy reference.

---

### MYLIB.CUSTS Access Descriptor

The MYLIB.CUSTS access descriptor for the CUSTOMERS table was created as follows:

```
proc access dbms=datacom;
  create mylib.custs.access;
```



```

user=demo;
table=customers;
assign = yes;
drop contact;
list all;
extend all;

rename customer = custnum telephone = phone
       streetaddress = street;
format firstorderdate = date7.;
informat firstorderdate = date7.;
content firstorderdate = yymmdd6.;
list all;
run;

```

---

## MYLIB.EMPLOYEE Access Descriptor

The MYLIB.EMPLOYEE access descriptor for the EMPLOYEEES table was created as follows:

```

proc access dbms=datacom;
create mylib.employee.access;
user=demo;
table=employees;
assign = yes;
format hiredate = date7.;
informat hiredate = date7.;
content hiredate = yymmdd6.;
format birthdate = date7.;
informat birthdate = date7.;
content birthdate = yymmdd6.;
list all;
extend all;
run;

```

---

## MYLIB.INVOICE Access Descriptor

The MYLIB.INVOICE access descriptor for the INVOICE table was created as follows:

```

proc access dbms=datacom;
create mylib.invoice.access;
user=demo;
table=invoice;
assign = yes;
format billedon = date7.;
informat billedon = date7.;
content billedon = yymmdd6.;
format paidon = date7.;
informat paidon = date7.;
content paidon = yymmdd6.;
list all;
extend all;
run;

```

---

## MYLIB.ORDER Access Descriptor

The MYLIB.ORDER access descriptor for the ORDER table was created as follows:

```
proc access dbms=datacom;
  create mylib.orders.access;
  user=demo;
  table=order;
  assign = yes;
  format dateordered = date7.;
  informat dateordered = date7.;
  content dateordered = yymmdd6.;
  format shipped = date7.;
  informat shipped = date7.;
  content shipped = yymmdd6.;
  list all;
  extend all;
run;
```

---

## View Descriptors for the CA-DATACOM/DB Tables

This section shows the view descriptors used in this manual to access data stored in the CA-DATACOM/DB tables. The tables are described in “CA-DATACOM/DB Tables” on page 126. The view descriptors are presented in alphabetical order for easy reference.

---

### VLIB.ALLEMP View Descriptor

The VLIB.ALLEMP view descriptor was created as follows:

```
proc access dbms=datacom ad=mylib.employee;
  create vlib.allemp.view;
  select all;
  list view;
run;
```

---

### VLIB.ALLORDR View Descriptor

The VLIB.ALLORDR view descriptor was created as follows:

```
proc access dbms=datacom ad=mylib.orders;
  create vlib.allordr.view;
  select all;
  list view;
run;
```

---

### VLIB.CUSORDR View Descriptor

The VLIB.CUSORDR view descriptor contains no selection criteria.

```
proc access dbms=datacom ad=mylib.orders;
  create vlib.cusordr.view;
```

```

select stocknum shipto;
list view;
run;

```

---

### **VLIB.CUSPHON View Descriptor**

The VLIB.CUSPHON view descriptor was created as follows:

```

proc access dbms=datacom ad=mylib.custs;
create vlib.cusphon.view;
select customer telephone name;
list view;
run;

```

---

### **VLIB.CUSTADD View Descriptor**

The VLIB.CUSTADD view descriptor was created as follows:

```

proc access dbms=datacom ad=mylib.custs;
create vlib.custadd.view;
select state zipcode country name city;
list view;
run;

```

---

### **VLIB.DCMEMPS View Descriptor**

The VLIB.DCMEMPS view descriptor was created as follows:

```

proc access dbms=datacom ad=mylib.employee;
create vlib.dcmemps.view;
select empid birthdate lastname firstname
middlename;
list view;
run;

```

---

### **VLIB.EMPINFO View Descriptor**

The VLIB.EMPINFO view descriptor was created as follows:

```

proc access dbms=datacom ad=mylib.employee;
create vlib.empinfo.view;
select empid dept lastname;
list view;
run;

```

---

### **VLIB.EMPS View Descriptor**

The VLIB.EMPS view descriptor was created as follows:

```

proc access dbms=datacom ad=mylib.employee;
create vlib.emps.view;
select empid jobcode birthdate lastname;
subset where jobcode = 602;

```

```

subset sort lastname;
list view;
run;

```

---

### **VLIB.FORINV View Descriptor**

The VLIB.FORINV view descriptor was created as follows:

```

proc access dbms=datacom ad=mylib.invoice;
create vlib.forinv.view;
select all;
subset where country != 'USA';
list view;
run

```

---

### **VLIB.INV View Descriptor**

The VLIB.INV view descriptor was created as follows:

```

proc access dbms=datacom ad=mylib.invoice;
create vlib.inv.view;
select invoicenum amtbilled country billedby paidon;
subset sort billedby;
list view;
run;

```

---

### **VLIB.USACUST View Descriptor**

The VLIB.USACUST view descriptor was created as follows:

```

proc access dbms=datacom ad=mylib.custs;
create vlib.usacust.view;
select customer state zipcode name
      firstorderdate;
list view;
extend view;

subset where customer eq 1#;
subset sort firstorderdate;
list view;

list all;
run;

```

---

### **VLIB.USAINV View Descriptor**

The VLIB.USAINV view descriptor was created as follows:

```

proc access dbms=datacom ad=mylib.invoice;
create vlib.usainv.view;
select all;
subset where country = 'USA';
list view;

```

```
run;
```

---

## **VLIB.USAORDR View Descriptor**

The VLIB.USAORDR view descriptor was created as follows:

```
proc access dbms=datacom ad=mylib.orders;
  create vlib.usaordr.view;
  select ordernum stocknum length fabriccharges shipto;
  subset sort shipto;
  subset where shipto = 1#;
  list view;
run;
```

---

## **SAS Data Files**

This section describes the SAS data files used in this book. It provides the SAS statements that create the data file and shows the output from the PRINT procedure.

---

### **MYDATA.OUTOFSTK Data File**

The SAS data file MYDATA.OUTOFSTK is used in Chapter 4, “Using CA-DATACOM/DB Data in SAS Programs,” on page 23. It was created with the following SAS statements:

```
libname mydata 'your-SAS-library';
data mydata.outofstk;
  input fibernam $8. fibernum;
  cards;
olefin 3478
gold 8934
dacron 4789
;
run;
```

The following PRINT procedure lists the output shown in Output A3.5 on page 146.

```
proc print data=mydata.outofstk;
  title 'SAS Data File MYDATA.OUTOFSTK';
run;
```

**Output A3.5** SAS Data File MYDATA.OUTOFSTK

SAS Data File MYDATA.OUTOFSTK		
OBS	FIBERNAM	FIBERNUM
1	olefin	3478
2	gold	8934
3	dacron	4789

**MYDATA.SASEMPS Data File**

The SAS data file MYDATA.SASEMPS is used in Chapter 5, “Browsing and Updating CA-DATACOM/DB Data,” on page 45. It was created with the following SAS statements:

```
libname mydata 'your-SAS-library';
data mydata.sasemps;
  input empid birthdat date7. lastname $18. firstnam $15.
        middlena $15.;
  cards;
245962 30AUG64 BEDORTHA           KATHY           MARTHA
765432 01MAR59 POWELL             FRANK           X.
219223 13JUN47 HANSINGER          BENJAMIN        HAROLD
326745 21FEB52 RAWN              BEATRICE        MAY
;
run;
```

The following PRINT procedure lists the output shown in Output A3.6 on page 146.

```
proc print data=mydata.sasemps;
  format birthdat date7.;
  title 'Data in MYDATA.SASEMPS Data File';
run;
```

**Output A3.6** SAS Data File MYDATA.SASEMPS

Data in MYDATA.SASEMPS Data File					
OBS	EMPID	BIRTHDAT	LASTNAME	FIRSTNAM	MIDDLENA
1	245962	30AUG64	BEDORTHA	KATHY	MARTHA
2	765432	01MAR59	POWELL	FRANK	X.
3	219223	13JUN47	HANSINGER	BENJAMIN	HAROLD
4	326745	21FEB52	RAWN	BEATRICE	MAY

## LIB6.BIRTHDAY Data File

The SAS data file LIB6.BIRTHDAY is used in Chapter 4, “Using CA-DATACOM/DB Data in SAS Programs,” on page 23. It was created with the following SAS statements:

```
libname lib6 'your-SAS-library';
data lib6.birthday;
  input empid birthdat date7. lastname $18.;
  cards;
129540 31JUL60 CHOULAI
356134 25OCT60 DUNNETT
127845 25DEC43 MEDER
677890 24APR65 NISHIMATSU-LYNCH
459287 05JAN34 RODRIGUES
346917 15MAR50 SHIEKELESLAN
254896 06APR49 TAYLOR-HUNYADI
;
run;
```

The following PRINT procedure lists the output shown in Output A3.7 on page 147.

```
proc print data=lib6.birthday;
  format birthdat date7.;
  title 'LIB6.BIRTHDAY Data File';
run;
```

**Output A3.7** SAS Data File LIB6.BIRTHDAY

LIB6.BIRTHDAY Data File			
OBS	EMPID	BIRTHDAT	LASTNAME
1	129540	31JUL60	CHOULAI
2	356134	25OCT60	DUNNETT
3	127845	25DEC43	MEDER
4	677890	24APR65	NISHIMATSU-LYNCH
5	459287	05JAN34	RODRIGUES
6	346917	15MAR50	SHIEKELESLAN
7	254896	06APR49	TAYLOR-HUNYADI





# Glossary

---

**access descriptor**

a SAS/ACCESS file that describes to the SAS System a single CA-DATACOM/DB table. It is used as a master for creating view descriptors. See also view descriptor.

**area**

an entity-type consisting of one or more files that hold the tables of a CA-DATACOM/DB database. Each AREA entity-occurrence requires a unique name and specified attributes in the CA-DATADITIONARY database.

**batch mode**

a method of executing SAS programs in which you prepare a file containing SAS statements, or job control language (JCL) statements, and any necessary operating system commands, and submit the program to the computer's batch queue. Batch execution is completely separate from other activities at your terminal. Batch mode is sometimes referred to as running in background.

**browsing data**

the process of viewing a file. You can see the data one row at a time or in a tabular format. You cannot update data that you are browsing.

**column**

a variable in a SAS data file. Each SAS column can have the following attributes: name, type (character or numeric), length, format, informat, and label. In the ACCESS procedure, columns are created from CA-DATACOM/DB fields.

**compound field**

a field containing two or more CA-DATACOM/DB fields that are contiguous. A compound field has a name and attributes in the CA-DATADITIONARY database. Each compound field allows access to a group of subfields as one unit. In contrast, see simple field.

**database**

an organized collection of related data. In CA-DATACOM/DB, a database is an entity-type that contains areas, files, records, and fields. Each DATABASE entity-occurrence has a name and attributes in the CA-DATADITIONARY database.

**data management system (DBMS)**

an integrated software package that enables you to create and manipulate data in the form of databases.

**CA-DATADICTIONARY database**

a blueprint for the type of data to be stored in a CA-DATACOM/DB database. The CA-DATADICTIONARY database consists of specifications for the various database, area, file, record, key, element, and field entity-occurrences in a database. CA-DATADICTIONARY stores and maintains the entity-occurrence names and attributes, along with keys and so on as needed by the application database.

**data type**

a classification according to the representation of the values to be stored in a CA-DATACOM/DB database. The data type is an attribute of every CA-DATACOM/DB field. It tells CA-DATACOM/DB how much physical storage to set aside for the field and the type of data the field will contain. CA-DATACOM/DB allows sixteen different types of character and numeric data. The CA-DATACOM/DB data type is similar to the type attribute of SAS columns.

**data value**

a character or numeric value that is stored in one column in a SAS row, that is, the intersection of a column (vertical component) and a row (horizontal component). It refers to the actual data in a SAS data file, such as the name Smith for the column LASTNAME.

In CA-DATACOM/DB, a character or numeric value that is stored in a field.

**descriptor files**

the SAS/ACCESS files used to establish a connection between the SAS System and CA-DATACOM/DB. To create descriptor files, you use the ACCESS procedure. There are two types of descriptor files, access descriptors and view descriptors.

**editing data**

the process of viewing a file with the intention (and ability) of changing it. You can see the data one record at a time or in a tabular format.

**element**

a CA-DATACOM/DB unit of transfer between application programs and CA-DATACOM/DB. An element consists of one or more contiguous CA-DATACOM/DB fields.

**engine**

a component of the SAS System that reads from or writes to a file. Each engine allows the SAS System to access files with a particular format. See also interface view engine.

**entity-occurrence**

an occurrence of a CA-DATACOM/DB entity-type, for example, a database, area, record, or field. Each entity-occurrence has a name and assigned attributes stored in the CA-DATADICTIONARY database.

**entity-type**

a classification of CA-DATACOM/DB entities that make up a CA-DATACOM/DB database. For example, an entity can be a database, area, record, or field.

**field**

a CA-DATACOM/DB entity-type that contains actual data values. Each FIELD entity-occurrence requires a unique name and specific attributes in the CA-DATADICTIONARY database. Records in a table contain one or more FIELD entity-occurrences. See also key field, simple field, compound field, and repeating field.

**file**

a collection of related records treated as a unit. SAS files are processed and controlled through the SAS System and are stored in a SAS data library.

In CA-DATACOM/DB, each database contains one or more FILE entity-occurrences that comprise specific records, fields, and elements. Each FILE entity-occurrence requires a unique name and specific attributes in the CA-DATADITIONARY database.

**format, column**

a set of directions the SAS System uses to display each character or numeric data value in a column.

**index**

a part of a SAS data file that stores both the values of a column and a set of directions. Under certain circumstances, using indexes enables the SAS System to locate rows in a data file more quickly and efficiently. Having columns indexed usually makes data set processing faster, although the SAS System determines the most efficient way to process data maintained by the SAS System.

In CA-DATACOM/DB, an Index Area is required for each database. Indexing allows any record in the database to be retrieved, according to the record's content. The index contains an entry for each key value in each record in the database. If a field is not indexed, values are not indexed, although they can be searched sequentially.

**informat, column**

a set of directions that the SAS System uses when reading column values.

**interactive line mode**

an execution mode in which program statements are entered on the terminal at the SAS session prompt. Procedure output and informative messages are returned directly to your monitor display screen.

**interface view engine**

a SAS System engine that retrieves data directly from files formatted by other software vendors. The SAS/ACCESS interface to CA-DATACOM/DB software includes an interface view engine.

**key**

a CA-DATACOM/DB component that contains one or more simple or compound fields. Keys identify a record or relate multiple records. You can also use keys to specify retrieval order. The fields within a key do not have to be contiguous. Each key must have a CA-DATACOM/DB ID in the CA-DATADITIONARY database. See also Native Key and Master Key.

**level**

in a CA-DATACOM/DB record, fields can be nested within each other as components, subcomponents, and so on. Each nested field has a nesting level.

**libref**

a temporary name that points to a SAS data library. A SAS file's complete name consists of two words separated by a period. The libref is the first word and indicates the library; the second word is the specific SAS file in the library. For example, in VLIB.NEWBDAY, VLIB is the libref and tells the SAS System where to look to find the file NEWBDAY.

**Native Key**

a field that determines the default sequence for the CA-DATACOM/DB data that are loaded and maintained in a CA-DATACOM/DB table. Each table must have one Native Key, and it can have only one. The Native Key can also be the Master Key.

**Master Key**

a field that makes CA-DATACOM/DB retrieval easier and faster. You can prevent the Master Key values from being duplicated or changed. The Master Key has a name

and specific attributes in the CA-DATADITIONARY database. Each table must have one Master Key, and it can have only one. The Master Key can also be the Native Key.

**member**

a SAS file in a SAS data library.

**member name**

a name given to a SAS file in a SAS data library.

**member type**

a SAS name that identifies the type of information stored in the file. Member types include access, data, catalog, program, and view.

**missing value**

a value in the SAS System indicating that no data are stored for the column in the current row. By default, the SAS System represents a missing numeric value with a single period and a missing character value by a blank space.

In a CA-DATACOM/DB database, a missing value contains all blanks.

**Multi-User environment**

a CA-DATACOM/DB execution environment in which several users access a CA-DATACOM/DB database at the same time, with queries and updates being handled simultaneously by a single copy of the software. See also Single User environment.

**password**

an optional string of characters that must be correctly specified by a user to gain access to a CA-DATADITIONARY entity-occurrence.

**record**

the horizontal component of a CA-DATACOM/DB table. A record is a set of fields that are treated as a unit. Records within a table are ordered by the Native Key. A record is analogous to a SAS row.

**repeating field**

a CA-DATACOM/DB field that occurs more than once. A repeating field can be a simple field or a compound field. Also, repeating fields can be nested within other repeating fields at different levels. Each repeating field has a name and specified attributes.

**row**

the horizontal component of a SAS data file. It is a collection of data values associated with a single entity, such as a customer or a state. Each row contains one data value for each column in the data file. A row is analogous to a data record in a CA-DATACOM/DB table.

**SAS data file**

one of the forms of a SAS data set implemented in Version 6 of the SAS System. A SAS data file contains both data values and the descriptor information associated with the data, such as the column attributes. In previous releases of the SAS System, all SAS data sets were SAS data files. SAS data files are of the member type data.

**SAS data library**

a collection of SAS data sets and other SAS files that are stored and referenced as a unit.

**SAS data set**

a collection of information stored as a unit under the SAS System. Several SAS data sets can be stored in a SAS data library. Unlike external files, a SAS data set is processed and controlled only through the SAS System.

A SAS data set is arranged in a rectangular two-dimensional format. Each item in a SAS data set is called a data value. Data values in a row make up an observation, and those in a column make up a variable. See also SAS data file and SAS data view for how SAS data sets are implemented in Version 6 of the SAS System.

**SAS data view**

one of the forms of a SAS data set implemented in Version 6 of the SAS System. A SAS data view contains only the descriptor and other information required to retrieve the data values from other SAS files or external files. Both PROC SQL views and SAS/ACCESS views are considered SAS data views. SAS data views are of the member type view.

**simple field**

a single CA-DATACOM/DB field that is not further divided. In contrast, see compound field.

**Single User environment**

a CA-DATACOM/DB environment that does not allow more than one user to share a database at the same time. See also Multi-User environment.

**status**

a value that identifies a copy of an entity-occurrence in a CA-DATACOM/DB database. The values for status are: TEST, PRODUCTION, HISTORY, INCOMPLETE, and QUALIFIED production, though only the first two are supported by the SAS/ACCESS interface to CA-DATACOM/DB. Typically, entities have a status consisting of the first letter followed by a three-character version number, for example, P001.

**table**

a collection of data records. In the CA-DATADITIONARY database, the combination of one FILE entity-occurrence and one RECORD entity-occurrence describes a table. In the SAS System's SQL procedure, a table is equivalent to a SAS data file.

**type**

See data type.

**URT**

See User Requirements Table.

**userid**

a string of characters that must be specified correctly by a user to gain access to CA-DATADITIONARY.

**User Requirements Table**

a table that specifies which CA-DATACOM/DB resources are required and allowed for programs that access CA-DATACOM/DB databases. A User Requirements Table (URT) must be associated with every program that requests CA-DATACOM/DB services. The URT defines, for example, the tables that the program can access (or update), processing priorities in a Multi-User environment, and transaction backout requirements.

**value**

See data value.

**view**

a definition of a virtual data set that is named and stored for later use. This file contains no data, but it describes or defines data stored elsewhere. SAS data views can be created by the ACCESS and SQL procedures. See also SAS data view.

For example, when a SAS/ACCESS view (a view descriptor) is referenced in the SAS PRINT procedure, the interface view engine reads data directly from a CA-DATACOM/DB table. You can also reference this view in certain SAS procedures to update the database described by a SAS/ACCESS view.

A view defined by the SQL procedure reads data from its underlying SAS data files, other PROC SQL views, or SAS/ACCESS views. Its output table can be a subset or a superset of one or more underlying structures. However, in Release 6.06, you cannot reference a PROC SQL view to update its underlying data.

Views specified with CA-DATADICTIONARY for CA-DATACOM/DB databases are not used by the SAS/ACCESS interface to CA-DATACOM/DB.

**view descriptor**

a SAS/ACCESS file that defines a subset of DBMS data described by an access descriptor. The subset consists of selected records in a given CA-DATACOM/DB table. See also access descriptor.

**windowing procedure**

a procedure that uses windows and menus to accomplish a SAS System task. For example, ACCESS and FSVIEW are windowing procedures.

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.