



CHAPTER

1

CA-OpenIngres Chapter, First Edition

<i>Introduction</i>	1
<i>SAS/ACCESS LIBNAME Statement</i>	2
<i>SAS/ACCESS Data Set Options: CA-OpenIngres Specifics</i>	5
<i>SAS System Option for the Interface to CA-OpenIngres</i>	7
<i>ACCESS Procedure: CA-OpenIngres Specifics</i>	8
<i>ACCESS Procedure Statements for CA-OpenIngres</i>	8
<i>DBLOAD Procedure: CA-OpenIngres Specifics</i>	10
<i>DBLOAD Procedure Statements for CA-OpenIngres</i>	10
<i>PROC DBLOAD Options</i>	10
<i>PROC DBLOAD Statements</i>	11
<i>Statement Interactions with the APPEND Option</i>	12
<i>SQL Procedure Pass-Through Facility: CA-OpenIngres Specifics</i>	13
<i>Arguments to Connect to CA-OpenIngres</i>	13
<i>CA-OpenIngres Naming Conventions</i>	14
<i>CA-OpenIngres SQL Data Types</i>	14
<i>Character Data</i>	15
<i>Numeric Data</i>	15
<i>Abstract Data</i>	16
<i>CA-OpenIngres NULL and Default Values</i>	18
<i>LIBNAME Statement Data Conversions</i>	18
<i>ACCESS Procedure Data Conversions</i>	19
<i>DBLOAD Procedure Data Conversions</i>	20
<i>Distributed CA-OpenIngres</i>	20
<i>Controlling Locks</i>	21
<i>Read Only Locks</i>	21
<i>UPDATE Locks</i>	21

Introduction

This chapter introduces SAS System users to CA-OpenIngres, a relational database management system (DBMS). It accompanies and should be used with *SAS/ACCESS Software for Relational Databases: Reference*(order #57204).*

This chapter describes the SAS/ACCESS LIBNAME and data set options that are specific to CA-OpenIngres. It then focuses on the terms and concepts that help you use the SAS/ACCESS interface to CA-OpenIngres. Next, it describes the CA-OpenIngres-specific arguments and statements that you use in the ACCESS and DBLOAD procedures and in the SQL procedure's CONNECT statement.

* Copyright © 1999 by SAS Institute Inc., Cary, NC, USA. All rights reserved.

For general information about database management systems, including information for the database administrator about how the SAS/ACCESS interfaces work, refer to Appendix 2, "DBMS Overview and Information for the Database Administrator". For more information about CA-OpenIngres, refer to your CA-OpenIngres documentation.

SAS/ACCESS LIBNAME Statement

This section describes the LIBNAME statement and its options that are specific to CA-OpenIngres. The LIBNAME statement and options that can be used in most databases are fully described in Chapter 3, "SAS/ACCESS LIBNAME Statement". This section also describes the connection options for CA-OpenIngres.

SAS/ACCESS LIBNAME Statement: CA-OpenIngres Specifics

Associates a SAS libref with a DBMS database, schema, server, or group of tables and views.

Valid: Anywhere

Syntax

```
LIBNAME libref SAS/ACCESS-engine-name SAS/ACCESS-engine-connection-options
      <SAS/ACCESS-LIBNAME-options>;
```

Arguments

libref

is any SAS name that serves as an alias to associate the SAS system with a database, schema, server, or group of tables and views.

SAS/ACCESS-engine-name

is a SAS/ACCESS engine name for your DBMS. The value in this argument is **INGRES**. SAS/ACCESS engines are implemented differently in different operating environments. The engine name is required.

SAS/ACCESS-engine-connection-options

are options that you specify in order to connect to a particular database; these options are different for each database. If the SAS/ACCESS engine connection options contain characters that are not allowed in SAS names, enclose the values of the options in quotation marks. On some DBMSs, if you specify the appropriate system options or environment variables for your database, you can often omit the SAS/ACCESS engine connection options.

SAS/ACCESS-LIBNAME-options

are options that apply to the objects in a DBMS, such as its tables or indexes. For example, the SPOOL= option enables you to specify whether or not SAS creates a utility spool file during read operations. Support for many of these options is DBMS specific.

Some SAS/ACCESS LIBNAME options can also be specified as SAS/ACCESS engine data set options. When you specify an option in the LIBNAME statement, it

applies to objects and data that are referenced by the libref. A SAS/ACCESS data set option applies only to the data set on which it is specified. If a like-named option is specified in both the SAS/ACCESS engine LIBNAME statement and after a data set name (which references a DBMS table or view), the SAS System uses the value that is specified after the data set name. Refer to Chapter 4, "SAS/ACCESS Data Set Options" for more information.

Details The LIBNAME statement associates a libref with a SAS/ACCESS engine in order to access tables or views in a database management system (DBMS). The SAS/ACCESS engine enables you to connect to a particular DBMS and, therefore, to specify a DBMS table or view name in a two-level SAS name.

For example, in MYLIB.EMPLOYEES_Q2, MYLIB is a SAS libref that points to a particular group of DBMS objects, and EMPLOYEES_Q2 is a DBMS table name. When you specify MYLIB.EMPLOYEES_Q2 in a DATA step or procedure, you dynamically access the DBMS table. Beginning in Version 7, SAS software supports reading, updating, creating, and deleting DBMS tables.

To disassociate or clear a libref from a DBMS, use a LIBNAME statement that specifies the libref (for example, MYDBLIB) and the CLEAR option as follows:

```
libname mydblib CLEAR;
```

The SAS/ACCESS engine disconnects from the database and closes any free threads or resources that are associated with that connection.

See for more information on options that you can use in the LIBNAME statement.

SAS/ACCESS Engine Connection Options The SAS/ACCESS engine connection options for CA-OpenIngres are as follows:

CONN_OPTIONS= on page 3

DATABASE= on page 3

PASSWORD= on page 4

SCHEMA= on page 4

USER= on page 4

CONN_OPTIONS=*"connection-options"*

specifies optional connection flags to the CA-OpenIngres DBMS. You can use any flag that is valid in CA-OpenIngres SQL. Connection options must be specified in quotation marks and separated by commas.

DATABASE= *'database-name'*

specifies the name of the CA-OpenIngres database that contains the tables and views that you want to access.

The DATABASE= option is required in order to connect to CA-OpenIngres. If you use a remote database, enter its remote virtual node name. If you use a distributed database, enter the database server type. If you use a remote distributed database, enter the database's virtual node name and server type. Quotation marks are required around the entire database string. The syntax for the database field follows:

```
<'v-node::>database</server—type'>
```

where

v-node is the name of the remote virtual node where the database is located. It implies that the network node address and protocol are defined by the NETU utility provided by CA-OpenIngres. Refer to your CA-OpenIngres administrator's guide for more information on the NETU utility.

database is the name of the database.

server-type is the type of server that is being accessed at the local or remote site. Valid server types are **INGRES** for a CA-OpenIngres DBMS server or **STAR** for a

CA-OpenIngres/STAR server. (You can replace **/STAR** with **/D**.) If you omit a server type, the default server type, INGRES, for the CA-OpenIngres DBMS, is used.

For example, to access the distributed database called TEXTILE on a remote virtual node named ATLANTA, you would enter 'ATLANTA::TEXTILE/STAR'.

PASSWORD=*password*

specifies the password that is associated with a username.

The **USER=** and **PASSWORD=** connection options are optional in CA-OpenIngres because users have pre-configured userids on their respective nodes.

SCHEMA=*schema-name*

enables you to specify which schema a libref references when you connect to CA-OpenIngres. When **SCHEMA=** refers to a userid that is different from your default userid, you must have the appropriate GRANT privileges to be able to access tables in that schema.

By default, when you connect to CA-OpenIngres, all tables are referenced by using your default schema.

USER=*username*

enables you to connect to a CA-OpenIngres database with a user ID that is different from the default ID.

The **USER=** and **PASSWORD=** connections are optional in CA-OpenIngres because users have pre-configured userids on their respective nodes. If **USER=** is omitted, your default userid is used.

SAS/ACCESS LIBNAME Options

This section describes the LIBNAME statement and its options that are specific to CA-OpenIngres. The SAS/ACCESS interface to CA-OpenIngres supports all of the SAS/ACCESS LIBNAME options listed in Chapter 3, "SAS/ACCESS LIBNAME Statement", except for **READ_LOCK_TYPE=** and **UPDATE_LOCK_TYPE=**. In addition to the supported options, the following LIBNAME options are used only in the interface to CA-OpenIngres or have CA-OpenIngres-specific aspects to them:

DBINDEX= on page 4

PRESERVE_COL_NAMES= on page 4

PRESERVE_TAB_NAMES = on page 4

SPOOL= on page 5

DBINDEX=YES|NO

indicates whether or not SAS calls CA-OpenIngres to find all indexes on the specified table.

The default value for CA-OpenIngres is NO. For a full description of this option, refer to Chapter 3, "SAS/ACCESS LIBNAME Statement".

PRESERVE_COL_NAMES=YES|NO

preserves spaces, special characters, and mixed case in CA-OpenIngres column names if the database was created with the ANSI/ISO Entry SQL-92 setting. Otherwise, **PRESERVE_COL_NAMES=** only preserves special characters and does not preserve case.

The default value for **PRESERVE_COL_NAMES=** for CA-OpenIngres is NO because CA-OpenIngres is a case-insensitive DBMS. For a full description of this option, refer to Chapter 3, "SAS/ACCESS LIBNAME Statement".

PRESERVE_TAB_NAMES=YES|NO

preserves spaces, special characters, and mixed case in CA-OpenIngres table names if the database was created with the ANSI/ISO Entry SQL-92 setting.

Otherwise, PRESERVE_TAB_NAMES= only preserves special characters and does not preserve case.

The default value for PRESERVE_TAB_NAMES= for CA-OpenIngres is NO because CA-OpenIngres is a case-insensitive DBMS. For a full description of this option, refer to Chapter 3, "SAS/ACCESS LIBNAME Statement".

SPOOL=YES|NO

specifies whether or not SAS creates a utility spool file during read operations that are performed with the specified LIBNAME.

If you omit SPOOL=, the default value is SPOOL=YES. For a full description of this option, refer to Chapter 3, "SAS/ACCESS LIBNAME Statement".

Example: Specifying a LIBNAME Statement to Access CA-OpenIngres Data

In this example, the libref MYLIB uses the CA-OpenIngres engine to connect to a CA-OpenIngres database. The SAS/ACCESS engine connection options are USER=, PASSWORD=, DATABASE=, CONN_OPTIONS=, and SCHEMA=.

```
libname mylib ingres
  user=tester1 password=myingpass
  database=hrdept
  conn_options="-xw"
  schema=levine;
proc print data=mylib.staff;
  where state='CA';
run;
```

See Also

The SQL Chapter in *SAS Procedures Guide*.

SAS/ACCESS Data Set Options: CA-OpenIngres Specifics

This section describes options that can be applied to SAS data sets that access data in CA-OpenIngres tables and views. In some cases, the option is fully described in "SAS/ACCESS Software for Relational Databases," except for some CA-OpenIngres-specific detail, such as a default value. In other cases, the entire option is CA-OpenIngres specific, so it is fully described in this chapter.

When specified in a DATA step or SAS procedure, the following data set options can be used on a SAS data set that accesses data in a DBMS object, such as a table or view. A data set option applies only to the SAS data set, or DBMS object, on which it is specified.

The SAS/ACCESS interface to CA-OpenIngres supports all of the SAS/ACCESS data set options listed in Chapter 4, "SAS/ACCESS Data Set Options", except for DBCREATE_TABLE_OPTS=, DBGEN_NAMES=, DBLABEL=, READ_LOCK_TYPE, and UPDATE_LOCK_TYPE=. In addition to the supported options, the following data set options are used only in the interface to CA-OpenIngres or have CA-OpenIngres-specific aspects to them:

DBINDEX= on page 6

DBNULL= on page 6

DBTYPE= on page 6

SASDATEFMT= on page 7

SCHEMA= on page 7

DBINDEX=

Indicates whether or not SAS calls CA-OpenIngres to find index(es) on the specified table.

Default value: NO

Syntax

DBINDEX=YES | NO | <'>*index-name*<'>

See Also

DBKEY=

For a full description of this option, refer to Chapter 4, "SAS/ACCESS Data Set Options".

DBNULL=

Indicates whether or not NULL is a valid value for the specified variables or columns.

Default value: NO

Syntax

DBNULL=(*variable1*=YES | NO *column1*=YES | NO)

For a full description of this option, refer to Chapter 4, "SAS/ACCESS Data Set Options".

DBTYPE=

Specifies whether or not to override the default CA-OpenIngres data type(s) when SAS outputs data to DBMS tables.

Default value: CHAR(), FLOAT

Syntax

DBTYPE=(*variable-name-1*=DBMS-type <...> <*variable-name-n*=DBMS-type>)

For a full description of this option, refer to Chapter 4, "SAS/ACCESS Data Set Options".

SASDATEFMT=

Changes the SAS date format or informat of a CA-OpenIngres column.

Default value: DATETIME16.

Syntax

SASDATEFMT=(DBMS-date-col='SAS-date-(in)format' ...)

SASDATEFMT= can also be specified with the SASDATEINFMT alias.

For a full description of this option, refer to Chapter 4, "SAS/ACCESS Data Set Options".

SCHEMA=

Enables you to read database tables and views in the specified schema.

Default value: None

Syntax

SCHEMA=*schema-name*

If you have READ access to tables and views in another schema, you can use the SCHEMA= data set option to browse them. SCHEMA= can also be specified with the OWNER= alias.

SAS System Option for the Interface to CA-OpenIngres

When you use PROC ACCESS and PROC DBLOAD, you can still use the INGOPTS= system option to pass certain flags to the DBMS to override the default action. To pass these flags to the LIBNAME engine, you must use the CONN_OPTIONS= LIBNAME option. For a description of CONN_OPTIONS=, refer to CONN_OPTIONS= on page 3.

By default, CA-OpenIngres ignores exceptions and does not issue errors or warnings when arithmetic exceptions occur. By using the INGOPTS= option, you can specify whether you want exceptions handled as errors or warnings. The SAS/ACCESS interface view engine for CA-OpenIngres only supports the arithmetic handling mode flag **-xk**, where **k** can either be an **f** (fatal error) or a **w** (warning).

INGOPTS="INGARITH=-xf"

handles exceptions as errors and the transaction is not completed.

INGOPTS="INGARITH=-xw"

handles exceptions as warnings and the transaction is completed. This is the default action.

The INGOPTS= option can be included in a SAS configuration file, submitted on the SAS command line, or submitted from within the SAS System. For more information about options that set the arithmetic handling modes, refer to your CA-OpenIngres SQL reference manual.

ACCESS Procedure: CA-OpenIngres Specifics

Chapter 9, "ACCESS Procedure Reference" describes the generic options and procedure statements that enable you to create access descriptors, view descriptors, and SAS data files from DBMS data. The following section describes the DBMS-specific statements that you use in the SAS/ACCESS interface to CA-OpenIngres.

ACCESS Procedure Statements for CA-OpenIngres

To create an access descriptor, you use a database-description statement in the PROC ACCESS step, DATABASE=. The database-description statement supplies database-specific information to the SAS System. This statement must immediately follow the CREATE or UPDATE statement that specifies the access descriptor to be created or updated.

Database-description statements are required only when you create access descriptors. Because CA-OpenIngres information is stored in an access descriptor, you do not need to repeat this information when you create view descriptors.

The syntax for other statements that you use to create an access descriptor is provided below. Refer to Chapter 9, "ACCESS Procedure Reference" for complete descriptions of these statements and their uses.

Note: The SAS/ACCESS interface to CA-OpenIngres does not use the following procedure statements: USER= and PASSWORD=. These statements are described in Chapter 9, "ACCESS Procedure Reference" . Δ

The SAS/ACCESS interface to CA-OpenIngres uses the following procedure statements in interactive line, noninteractive, or batch mode:

```
PROC ACCESS <DBMS=INGRES | view-descriptor-options>;
CREATE libref.member-name.ACCESS | VIEW;
UPDATE libref.member-name.ACCESS | VIEW;
DATABASE=<'> <v-node::>>database</server-type><'>;
TABLE=<'>table-name<'>;
ASSIGN <=> YES | NO ;
DROP <'>column-identifier-1<'> <...<'>column-identifier-n<'>>;
FORMAT <'>column-identifier-1<'><=>SAS-format-name-1
    <...<'>column-identifier-n<'> <=> SAS-format-name-n>;
QUIT;
RENAME <'>column-identifier-1<'> <=> SAS-variable-name-1
    <...<'>column-identifier-n<'> <=> SAS-variable-name-n>;
```



```

RESET ALL | <'>column-identifier-1<'><...<'>column-identifier-n<'>>
SELECT ALL | <'>column-identifier-1<'> <...<'>column-identifier-n<'>>
SUBSET selection-criteria;
UNIQUE <=> YES | NO;
LIST <ALL | VIEW | <'>column-identifier<>>

```

RUN;

```

DATABASE=<'><v-node::>database</server-type><'>;

```

The DATABASE= statement specifies the name of the CA-OpenIngres database. This statement is required.

If you use a remote database, you enter its remote virtual node name. If you use a distributed database, you enter the database server type. If you use a remote distributed database, you enter the database virtual node name and server type. The syntax for the database field follows; quotes are required around the entire database string:

```

< 'v-node::>database</server-type'>

```

where

v-node is the name of the remote virtual node where the database is located. It implies that the network node address and protocol are defined by the NETU utility that is provided by CA-OpenIngres. For more information about the NETU utility, refer to your CA-OpenIngres/Net user's and administrator's guide.

database is the name of the database.

server-type is the type of server that is being accessed at the local or remote site. Valid server types are **INGRES** for a CA-OpenIngres DBMS server or **STAR** for a CA-OpenIngres/Star server. (You can replace **/STAR** with its alias, **/D**.) If you omit a server type, the default server type, **INGRES**, for the CA-OpenIngres DBMS is used.

For example, to access the distributed database called TEXTILE on a remote virtual node named ATLANTA, you would enter '**ATLANTA::TEXTILE/STAR**' .

```

proc access dbms=ingres;
  /*create access descriptor */
  create adlib.customer.access;
  database='testnode::testdatabase/star';
  table=customers;
  assign=yes;
  rename customer = custnum;
  format firstorder date9.;
  list all;

  /*create usacust view      */
  create vlib.usacust.view;
  select customer state zipcode name
         firstorder;
  subset where customer like '1%';
run;

```

DBLOAD Procedure: CA-OpenIngres Specifics

Chapter 10, "DBLOAD Procedure Reference" describes the generic options and procedure statements that enable you to create a DBMS table and to insert data in it. The following section describes the DBMS-specific statements that you use in the SAS/ACCESS interface to CA-OpenIngres.

DBLOAD Procedure Statements for CA-OpenIngres

To create and load a CA-OpenIngres table, use the DBMS=INGRES option and one database-description statement in the PROC DBLOAD step, DATABASE=. The database-description statement supplies database-specific information to the SAS System.

The syntax for other statements that you use to create and load a CA-OpenIngres table is provided below. Refer to Chapter 10, "DBLOAD Procedure Reference" for complete descriptions of these statements and their uses.

Note: The SAS/ACCESS interface to CA-OpenIngres DBLOAD procedure does not use the following procedure statements: USER= and PASSWORD=. These statements are described in Chapter 10, "DBLOAD Procedure Reference" . \triangle

The SAS/ACCESS interface to CA-OpenIngres DBLOAD procedure uses the following statements in interactive line, noninteractive, or batch mode:

```
PROC DBLOAD <DBMS=INGRES> <DATA=< libref.> SAS-data-set> <APPEND>;
  DATABASE=<' v-node::>database</server-type>;
  TABLE=<'>table-name<'>;
  ACCDESC= < libref.>access-descriptor;
  COMMIT=commit-frequency;
  DELETE variable-identifier-1 <...> variable-identifier-n;
  ERRLIMIT=error-limit;
  LABEL;
  LIMIT=load-limit;
  NULLS variable-identifier-1 Y|N <...variable-identifier-n Y|N>;
  QUIT;
  RENAME variable-identifier-1=<'>column-name-1<'>
    <...variable-identifier-n=<'>column-name-n<'>>;
  RESET ALL| variable-identifier-1< ...variable-identifier-n>;
  SQL CA-OpenIngres-SQL-statement;
  TYPE variable-identifier-1 = 'column-type-1'
    <...variable-identifier-n = 'column-type-n'>
  WHERE SAS-where-expression;
  LIST <list-selection>;
  LOAD;
RUN;
```

PROC DBLOAD Options

APPEND

enables you to load SAS data set observations into an existing CA-OpenIngres table, beginning with Release 6.12 of the SAS System. Any SAS data set (SAS data file or SAS data view) can be used as input for APPEND processing. APPEND is optional and can be specified only in the PROC DBLOAD step.

The APPEND option assumes that the table that is specified in the TABLE= statement exists. To create a CA-OpenIngres table to append to, first use PROC DBLOAD with the SQL statement. Then specify another PROC DBLOAD step with APPEND. If the table does not exist, the procedure fails and an error message is displayed.

Other PROC DBLOAD statements such as DELETE and RENAME work differently when they are used with the APPEND option.

Suppose that you have a data set named INVDATA that you want to append to an existing CA-OpenIngres table. The following example shows how to use the APPEND option to append rows from the INVDATA data set to the table named INVOICE.

```
proc dbload dbms=ingres data=invdata append;
  database='testnode::testdatabase/star';
  table=invoice; load;
run;
```

PROC DBLOAD Statements

DATABASE='<v-node::>database </server-type >';

indicates the name of the database where you want to store the new table. You must enclose it in single or double quotes. This statement is required. If you specify a database, it must exist. If you do not know the names of your databases, contact your DBA.

If you use a remote database, you enter the remote virtual node name. If you use a distributed database, you enter the database server type. If you use a remote distributed database, you enter the database node name and server type.

v-node

is the name of the remote virtual node where the database is located. It implies that the network node address and protocol have been defined by the NETU utility that is provided by CA-OpenIngres. If you are accessing a local database, you do not need to enter the virtual node name. For more information about the NETU utility, refer to your CA-OpenIngres/Net user's and administrator's guide.

database

is the name of the database.

server-type

is the type of server that is being accessed at the local or remote site. Valid server types are **INGRES** for a CA-OpenIngres DBMS server or **STAR** for a CA-OpenIngres/Star server. (You can replace **/STAR** with its alias, **/D**.) If you omit a server type, the default server type, **INGRES**, for the CA-OpenIngres DBMS is used.

The following example creates a new CA-OpenIngres table, INVOICE, from the DLIB.INVOICE data file. An access descriptor ADLIB.INVOICE is also created, based on the new table. The AMTBILLED column and the 5th column in the table (AMOUNTINUS) are renamed. You must be granted the appropriate privileges in order to create new CA-OpenIngres tables or views.

```

libname adlib 'SAS-data-library';
libname dlib 'SAS-data-library';

proc dbload dbms=ingres data=dlib.invoice;
  database='textiledb'; table=invoice;
  accdesc=adlib.invoice;
  rename amtbilled = amountbilled
         5 = amountindollars;
  nulls invnum=n amtbill=n;
  load;
run;

```

The next example only sends an SQL GRANT statement to the TEXTILEDB database and does not create a new table. Therefore, the TABLE= and LOAD statements are omitted.

```

proc dbload dbms=ingres;
  database='textiledb';
  sql grant select on customers to frederick;
run;

```

Statement Interactions with the APPEND Option

DELETE and RENAME

DELETE and RENAME enable you to perform one-to-one matching of SAS variables to CA-OpenIngres columns. When you use the APPEND option, there must be a CA-OpenIngres column for every SAS variable that you are loading.

Use the DELETE statement to drop a SAS input variable that has no corresponding CA-OpenIngres column. Use the RENAME statement if the names do not match. If there is not a CA-OpenIngres column for each SAS variable, the APPEND operation is rejected.

The RENAME statement indicates the column name in the CA-OpenIngres table for the SAS data set variable that you specify. This example loads the data that is associated with the SAS variable COUNTRY into the CA-OpenIngres column named ORIGIN.

```

rename country=origin;

```

ERRLIMIT=

The DBLOAD ERRLIMIT= statement has a default of 10 when the APPEND option is specified, instead of the default of 100.

LOAD

The LOAD statement is required.

NULLS

The NULLS statement has no effect when the APPEND option is specified, unless the table is created in the same step.

TYPE

The TYPE statement has no effect because the input SAS variable types and the CA-OpenIngres column types must have a one-to-one match.

SQL Procedure Pass-Through Facility: CA-OpenIngres Specifics

Chapter 6, "SQL Procedure's Interaction with SAS/ACCESS Software" describes the generic PROC SQL statements that you use to connect to and disconnect from a DBMS, to send DBMS-specific statements to the DBMS, and to retrieve DBMS data for your SAS programs. The following section describes the DBMS-specific arguments that you use in the CONNECT statement "Arguments to Connect to CA-OpenIngres" on page 13.

Arguments to Connect to CA-OpenIngres

CA-OpenIngres has one *database-connection-argument* that you can specify in this statement. It can also be used to connect to multiple databases.

```
CONNECT TO INGRES <AS alias> (DATABASE=< v-node::>database-name</  
  server-type>);
```

```
(DATABASE='< v-node::>database-name </server-type>')
```

is where you specify the name of the CA-OpenIngres database. You must enter a value in this field and enclose the entire option in parentheses.

If you use a remote database, you enter the remote virtual node name. If you use a distributed database, you enter the database server type. If you use a remote distributed database, you enter the remote virtual node name and server type. The arguments for the DATABASE= field are

v-node is the name of the remote virtual node where the database is located. It implies that the network node address and protocol have been defined by the NETU utility that is provided by CA-OpenIngres. For more information on the NETU utility, refer to your CA-OpenIngres/Net user's and administrator's guide.

database-name is the name of the database.

server-type is the type of server that is being accessed at the local or remote site. Valid server types are **INGRES** for a CA-OpenIngres DBMS server or **STAR** for the CA-OpenIngres/Star server. (You can replace **/STAR** with its alias, **/D**.) If you omit a server type, the default server type, INGRES, for the CA-OpenIngres DBMS is used.

The following example connects to CA-OpenIngres by using the alias ING1 and sends an EXECUTE statement. The distributed database name is TEXTILEDB, and it is located on a virtual node that is named TXTNODE. The alias D is used to specify the STAR server.

```
proc sql;
  connect to ingres as ing1
    (database='txtnode::textiledb/d');
  execute (create view whotookorders as
    select ordernum, takenby,
           firstname, lastname, phone
    from orders, employees
    where
      orders.takenby=employees.empid)
  by ing1;
```

```

execute (grant select on whotookorders
to adeleke) by ing1;
disconnect from ing1;
quit;

```

The following example connects to the TEXTILEDDB database and performs a query, shown in italics, on the CUSTOMERS table:

```

proc sql;
connect to ingres (database=textiledb);
select *
from connection to ingres
(select * from customers
where customer like '1%');
disconnect from ingres;
quit;

```

CA-OpenIngres Naming Conventions

CA-OpenIngres database objects that can be named include tables, views, columns, indexes, and database procedures. Use the following CA-OpenIngres naming conventions:

- A *database* name cannot begin with an underscore.
- A name starts with a letter or an underscore. Names cannot begin with numbers or special characters such as #, @, \$.
- A name can be from 1 to 24 characters long. Objects that are managed by the CA-OpenIngres user interfaces (QBF, RBF, ABF, and Vifred) can be up to 32 characters.
- A name can contain the letters A through Z, the digits 0 through 9, the underscore (`_`), the dollar sign (`$`), the pound sign (`#`), and the at sign (`@`). (Although dollar signs are permitted in names, they are also used in some operating system-supplied names. To avoid conflicts with these names, it is recommended that you not use dollar signs in your table, view, column, or index names.)
- Case significance is determined by the setting with which the database is created: CA-OpenIngres or ANSI/ISO Entry SQL-92-compliant. The default for CA-OpenIngres is lowercase, and the default for ANSI/ISO Entry SQL-92-compliant is uppercase.
- A table name cannot begin with `ii` or `sq`, which are reserved table names for use by CA-OpenIngres.
- A name may not be the same as another CA-OpenIngres database object that has the same type.

CA-OpenIngres SQL Data Types

Every column in a table has a name and a data type. The data type tells CA-OpenIngres how much physical storage to set aside for the column and the form in which the data is stored.

Character Data

The character data types for CA-OpenIngres SQL are listed below, where n is the length of the character string.

CHAR(n), C n

is the fixed-length character string where n represents a string of 1 to 2000 ASCII characters. CHAR is preferred over C. CHAR can contain any printing or nonprinting characters and the NULL character. When comparing CHAR strings, blanks are significant.

For compatibility with previous versions of CA-OpenIngres, C is supported. C can contain only printing characters; nonprinting characters are converted to blanks. When comparing C strings, blanks are ignored.

VARCHAR(n) TEXT(n)

is the varying-length character string where n represents a string of 1 to 2000 ASCII characters. VARCHAR is preferred over TEXT. VARCHAR can contain any printing and nonprinting characters and the NULL character. When comparing VARCHAR strings, blanks are significant. VARCHAR, when comparing strings of unequal length, adds blanks to the end of the shorter string to make it the same length as the longer string.

TEXT can contain all ASCII characters except NULL characters, which are converted to blanks. When comparing TEXT strings, blanks are significant. TEXT, when comparing strings of unequal length, considers a shorter string "less than" a longer string if all characters up to the length of the shorter string are equal.

LONG VARCHAR

is the varying-length column for character string data. A LONG VARCHAR string can be up to two gigabytes in length. SAS/ACCESS does not currently allow you to use data in CA-OpenIngres LONG VARCHAR columns.

TEXT

is the varying-length character string that can contain all ASCII characters except NULL characters. NULL characters are converted to blanks. When comparing TEXT strings, blanks are significant. TEXT, when comparing strings of unequal length, considers a shorter string "less than" a longer string if all characters up to the length of the shorter string are equal. TEXT is supported for compatibility with previous versions of CA-OpenIngres.

Numeric Data

The numeric data types for CA-OpenIngres SQL are listed below.

REAL | FLOAT4, DOUBLE PRECISION | FLOAT | FLOAT8

Floating-point values consist of the following: an integer part, a decimal point, and a fraction part or scientific notation of the format that follows:

$$\backslash[+[minus]\backslash\langle digit \rangle\backslash[. [lt]digit \rangle\backslasheE\backslash+[minus]\backslash\langle digit \rangle$$

where the clauses that are enclosed in the angle brackets (<>) are optional and, if used, can be repeated. The FLOAT4 (4 byte) range is $-1.0e+38$ to $+1.0e+38$ with 7-digit precision. The FLOAT (8 byte) range is $-1.0e+38$ to $+1.0e+38$ with 16-digit precision.

The default character that indicates the decimal point is a period (.). This character can be changed by setting II_DECIMAL. For more information on setting II_DECIMAL, refer to your CA-OpenIngres installation and operations guide.

Note: REAL is a synonym for FLOAT4. FLOAT8 and DOUBLE PRECISION are synonyms for FLOAT. Δ

SMALLINT | INTEGER2, INTEGER1, INTEGER | INTEGER4

Integers contain no fractional part. The three integer data types are INTEGER1 (1 byte), which has a range of -128 to $+127$, SMALLINT (2 byte), which has a range of $-32,768$ to $+32,767$, and INTEGER (4 byte), which has a range of $-2,147,483,648$ to $+2,147,483,647$.

Note: INTEGER2 is a synonym for SMALLINT. INTEGER4 is a synonym for INTEGER. Δ

DECIMAL | DEC | NUMERIC

exact numeric data type defined in terms of its precision (total number of digits) and scale (number of digits to the right of the decimal point). The minimum precision for a decimal value is 1 and the maximum is 31. The scale of a decimal value cannot exceed its precision. Scale can be 0 (no digits to the right of the decimal point).

Abstract Data

CA-OpenIngres SQL date and money data types are abstract data types.

DATE

Dates are input as quoted character strings. CA-OpenIngres accepts U.S. date input formats, absolute time input formats, and combined date and time input formats. DATE (12 bytes) has a range of 1-JAN-1582 to 31-DEC-2382 (for absolute dates) and -800 years to 800 years (for time intervals).

CA-OpenIngres assumes that the year is the current year if the year is omitted. If a two-digit year is used in formats with delimiters such as forward slashes (/) or dashes (-), or if the year is omitted from the date, CA-OpenIngres assumes that the century is 1900.

CA-OpenIngres assumes the current (today's) date if the date is omitted from an absolute time entry.

Dates are stored in Greenwich Mean Time (GMT). The SAS/ACCESS interface to CA-OpenIngres takes care of converting dates. In the following tables, *xxx* represents a time zone designation. Valid time zone designations are edt, cdt, mdt, pdt, gmt, est, cst, mst, and pst. CA-OpenIngres assumes the local time zone if a time zone designation is not supplied.

At this time, the SAS/ACCESS interface to CA-OpenIngres does not support the CA-OpenIngres date and time interval formats.

The date output format for CA-OpenIngres is FINLAND. You cannot change this format while you are using the SAS/ACCESS engine for Ingres. The FINLAND date output format allows the *yyyy-mm-dd* input format, plus all of the U.S. input formats except *mm-dd-yyyy*. (See Table 1.1 on page 17). For more information, refer to your CA-OpenIngres SQL reference manual and your CA-OpenIngres installation and operations guide.

For information about changing the date format conventions to accommodate international conventions, refer to your CA-OpenIngres SQL reference manual.

Table 1.1 Input Formats Allowed by FINLAND Date Input Format

Format	Example
'mm/dd/yyyy'	'05/16/1991'
'dd-mmm-yyyy'	'16-may-1991'
'yyyy.mm.dd'	'1991.05.16'
'yyyy_mm_dd'	'1991_05_16'
'yyyy-mm-dd'	'1991-05-16'
'mmdyy'	'051691'
'mm/dd'	'05/16'
'mm-dd'	'05-16'
'today'	The string ' <i>today</i> ' has today's date as its value.
'now'	The string ' <i>now</i> ' has the current date and time as its value

Table 1.2 Absolute Time Input Formats

Format	Example
'hh:mm:ss'	'10:30:00'
'hh:mm:ss xxx'	'10:30:00 est'
'hh:mm:ss [am pm]xxx'	'10:30:00 am est'
'hh:mm'	'10:30'
'hh:mm xxx'	'10:30 est'

Table 1.3 Combined Date and Time Input Formats

Format	Example
'mm/dd/yy hh:mm:ss'	'05/16/91 10:30:00'
'dd-mmm-yy hh:mm:ss'	'16-may-91 10:30:00'
'mm/dd/yy hh:mm:ss xxx'	'05/16/91 10:30:00 est'
'dd-mmm-yy hh:mm:ss xxx'	'16-may-91 10:30:00 est'
'dd-mmm-yy hh:mm:ss[am pm]xxx'	'05-may-91 10:30:00 am est'
'mm/dd/yy hh:mm'	'01/16/91 10:30'
'dd-mmm-yy hh:mm'	'16-may-91 10:30'

Format	Example
'mm/dd/yy hh:mm xxx'	'05/16/91 10:30 est'
'dd-mmm-yy hh:mm xxx'	'05-may-91 10:30 est'

MONEY

Money values in CA-OpenIngres are stored as their actual money amount. Money values are significant to exactly two decimal places, and on input and output their amounts are rounded in dollars and cents. Arithmetic operations on MONEY retain the precision of two decimal places. MONEY (8 byte) has a range of \$-999,999,999,999.99 to \$999,999,999,999.99.

Money values in CA-OpenIngres are displayed as 20-character strings. The display format is *\$sddddddddd.dd* where *s* is the sign (-for negative and no sign for positive), and *d* is a digit from 0 to 9.

CA-OpenIngres accepts money values on input as character strings, numbers, valid integers, or floating-point numbers.

For information about the logical names that affect the display of MONEY values, refer to your CA-OpenIngres installation and operations guide.

CA-OpenIngres NULL and Default Values

CA-OpenIngres has a special value that is called NULL. *NULL* means an absence of information and is analogous to the SAS System's *missing value*.

Columns can be defined as NOT NULL so that they cannot contain NULL data. NOT NULL tells CA-OpenIngres not to add a row to the table unless the row has a value for the column.

Columns can be defined so that they cannot contain NULL data. For example, the CREATE TABLE statement for the CUSTOMERS table in Appendix 1 defines the first column, CUSTOMER, as CHAR(8) and NOT NULL. NOT NULL tells CA-OpenIngres not to add a row to the table unless the row has a value for CUSTOMER.

LIBNAME Statement Data Conversions

The following table shows the default SAS System variable formats that the LIBNAME statement assigns to CA-OpenIngres data types during input operations.

Table 1.4 LIBNAME Statement: Default SAS Formats for CA-OpenIngres Data Types

INGRES Data Type	Default SAS Format
CHAR(<i>n</i>) or <i>Cn</i>	<i>\$n</i> . (<i>n</i> < 200) \$200. (<i>n</i> > = 200)
VARCHAR(<i>n</i>) or TEXT(<i>n</i>)	<i>\$n</i> . (<i>n</i> < 200) \$200. (<i>n</i> > = 200)
INTEGER1	4.0
SMALLINT (INTEGER2)	6.0
INTEGER (INTEGER4)	11.0
FLOAT4 (REAL)	none

INGRES Data Type	Default SAS Format
FLOAT (FLOAT8)	none
DATE	DATETIME21.
MONEY	DOLLAR19.2
DATE INTERVAL	NOT SUPPORTED
TIME INTERVAL	NOT SUPPORTED
DECIMAL	<i>m.n</i> ($m \leq 31$ and $n \leq m$)

The following table shows the default CA-OpenIngres data types that the LIBNAME statement assigns to SAS variable formats during output operations.

Table 1.5 LIBNAME Statement: Default CA-OpenIngres Data Types for SAS Variable Formats

SAS Variable Format	CA-OpenIngres Data Type
$\$w$	CHAR(<i>n</i>)
any date format	DATE
<i>w</i> .0 ($w \leq 2$)	INTEGER1
<i>w</i> .0 ($w > 2$ and $w \leq 4$)	SMALLINT
<i>w</i> .0 (where $w > 4$ and $w < 9$)	INTEGER
IB <i>w.d</i> , PIB <i>w.d</i>	INTEGER
all other numeric datatypes	FLOAT
DOLLAR	MONEY

ACCESS Procedure Data Conversions

The following table lists the default SAS System variable formats that the ACCESS procedure assigns to each CA-OpenIngres data type.

Table 1.6 PROC ACCESS: Default SAS Formats for CA-OpenIngres Data Types

INGRES Data Type	SAS Variable Format
CHAR(<i>n</i>) or C <i>n</i>	$\$n$. ($n < 200$) $\$200$. ($n \geq 200$)
VARCHAR(<i>n</i>) or TEXT(<i>n</i>)	$\$n$. ($n < 200$) $\$200$. ($n \geq 200$)
INTEGER1	4.0
SMALLINT (INTEGER2)	6.0
INTEGER (INTEGER4)	11.0

INGRES Data Type	SAS Variable Format
FLOAT4 (REAL)	E14.0
FLOAT (FLOAT8)	E23.0
DATE	DATETIME21.
MONEY	DOLLAR19.2
DATE INTERVAL	NOT SUPPORTED
TIME INTERVAL	NOT SUPPORTED

DBLOAD Procedure Data Conversions

The following table lists the default CA-OpenIngres data type that the DBLOAD procedure assigns to each SAS variable format.

Table 1.7 PROC DBLOAD: Default CA-OpenIngres Data Type for SAS Variable Formats

SAS Variable Format	INGRES Data Type
w.0	INTEGER1
w.0	SMALLINT
w.0	INTEGER
IBw.d, PIBw.d	INTEGER
all other numeric data types	FLOAT
DOLLAR	MONEY
any SAS date, datetime, or time format	DATE
\$n	CHAR(<i>n</i>)

Distributed CA-OpenIngres

The following products are used in the CA-OpenIngres distributed relational database management system to enable you to simultaneously access multiple CA-OpenIngres and non-CA-OpenIngres databases on local and remote servers.

CA-OpenIngres

is the CA-OpenIngres relational database management system.

CA-OpenIngres/Net

enables you to access CA-OpenIngres data on a remote server from your local computer through a link that uses an existing network.

CA-OpenIngres Gateways

enables you to access non-CA-OpenIngres data in other database servers and file management systems.

CA-OpenIngres /Star

enables you to access multiple local, remote, CA-OpenIngres, and non-CA-OpenIngres database servers at the same time.

To access the various databases of distributed CA-OpenIngres through SAS/ACCESS to CA-OpenIngres, you indicate, as appropriate, the virtual node name, the database name, and the type of server that is being accessed when you identify the database that is being used.

Refer to your CA-OpenIngres documentation for additional information.

Controlling Locks

Locks are completely under the control of CA-OpenIngres. The interface view engine, ACCESS procedure, and DBLOAD procedure all obtain locks of varying types on CA-OpenIngres tables and views.

Read Only Locks

When the interface view engine reads data from a CA-OpenIngres table or view, an SQL SELECT statement is issued to place a shared lock on the table. (A shared lock enables more than one user to access the data.) This lock is active for the entire execution of the procedure, regardless of the number of passes through the data the procedure requires. For example, the PRINT procedure places a shared lock on the table, and the lock remains active until all the specified data have been written to the Output window (that is, displayed).

UPDATE Locks

When you use the DATA step or certain procedures (such as FSEDIT or SQL) to update CA-OpenIngres data, an exclusive lock is placed on the CA-OpenIngres data. (An exclusive lock enables only a single user to access the data.) Except in the case of the DBLOAD procedure, these locks remain active only for the duration of a specific update and not for the entire procedure.

For example, in the FSEDIT procedure, if you update, delete, or add an observation, the interface starts an exclusive lock. That lock remains active until you complete your update, delete, or add.

Note: At every update, it is possible that a particular update process (for example, a PROC FSEDIT DATA=... statement) is not executed immediately. This indicates that another user has that CA-OpenIngres table locked. Your process remains in a "wait" state until the lock is released. If the "wait" is too long and times out, an error message is written to the SAS log. Δ

When a table is to be loaded, the DBLOAD procedure issues a CREATE table statement, which causes an exclusive lock to be placed on the table. This lock is active until a commit is issued. (If the table already exists, an error message is written to the SAS log and the new table is not created.)

When you use a PROC DBLOAD SQL statement or the Query Entry window to submit SQL statements to CA-OpenIngres, locks are obtained in the same way as if you had issued the SQL command from the CA-OpenIngres Interactive SQL utility (called the CA-OpenIngres terminal monitor).

The following table shows examples of the locks CA-OpenIngres issues for some of the SAS/FSP interactive windowing procedures.

Table 1.8 CA-OpenIngres Locks for Interactive SAS/FSP Procedures

Procedure	Lock Level	Lock
FSEEDIT	Table	IX (intended exclusive)
Page lock(s) on pages in table	X (exclusive or write)	
FSBROWSE	Table	IS (intended shared)
Page lock(s) on pages in table	S (shared or read)	
FSVIEW		
<input type="checkbox"/> for browsing	Table	IS (intended shared)
Page lock(s) on pages in table	S (shared or read)	
<input type="checkbox"/> for editing	Table	IX (intended exclusive)
	Page lock(s) on pages in table	X (exclusive or write)

To prevent users from being locked out of the data, you should encourage SAS/FSP software users to browse CA-OpenIngres data if they do not intend to change the data. You might also want to restrict who can use these procedures to access CA-OpenIngres data. For more information about locks, refer to your CA-OpenIngres database administrator's guide.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ACCESS® Software for Relational Databases: Reference, Version 8 (CA-OpenIngres Chapter)*, Cary, NC: SAS Institute Inc., 1999.

**SAS/ACCESS® Software for Relational Databases: Reference, Version 8
(CA-OpenIngres Chapter)**

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

ISBN 1-58025-535-3

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, 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.