



APPENDIX

4

CV2ODBC Procedure

<i>Overview</i>	263
<i>CV2ODBC Procedure Syntax</i>	263
<i>PROC CV2ODBC</i>	264
<i>FROM VIEW Statement</i>	264
<i>TO VIEW Statement</i>	264
<i>DSN Statement</i>	265
<i>UID Statement</i>	265
<i>PASSWORD Statement</i>	265
<i>SAVE Statement</i>	266
<i>SUBMIT Statement</i>	266
<i>REPLACE Statement</i>	267
<i>Examples</i>	267

Overview

The CV2ODBC procedure is used to convert Version 6 view descriptors for the AS/400 or MS SQL Server to Version 8 ODBC SQL views for Windows.

The conversion of PROC SQL Pass-Through views is automatic and does not require you to use the CV2ODBC procedure. See for more information.

CV2ODBC Procedure Syntax

```

PROC CV2ODBC DBMS=AS400 | MSSQL;
FROM <VIEW>=V6-view-descriptor <(PW=password)>;
TO <VIEW>=V8-view-name;
DSN=data-source-name;
UID | USER=user-id;
PWD | PASSWORD | PW =password;
SAVE <AS=>filename;
SUBMIT;
REPLACE ALL | VIEW | FILE;

```

PROC CV2ODBC

PROC CV2ODBC DBMS=AS400 | MSSQL;

Options

DBMS= AS400 | MSSQL

specifies the DBMS from which the view is being converted. Conversion is supported for the IBM AS/400 and Microsoft SQL Server. DBMS= is required.

FROM VIEW Statement

Specifies the view descriptor to convert.

FROM <VIEW>=*V6-view-descriptor* <(PW=*password*)>;

Arguments

V6-view-descriptor-name

specifies the existing Version 6 PROC ACCESS view descriptor that is to be converted. FROM VIEW= is required.

password

If the view was created with an optional SAS password, then the PW= statement is required. A SAS password is used for accessing a SAS view file. If a Version 6 view was created with a SAS password, then the password will always be associated with that view. Therefore, the converted Version 8 view will require the SAS password for any procedures that it uses.

TO VIEW Statement

Specifies the new view name.

TO <VIEW>=*V8-view-name*;

Arguments

V8-view-name

specifies the new Version 8 view name that you want to create. TO VIEW= is required. If PW= is specified in the FROM= statement, then it is automatically applied to this view and you will have to specify PW= for this statement also. If you do not specify PW=, then you will be prompted for it when the view is processed within a DATA step or another procedure.

DSN Statement

Specifies the data source name.

DSN=*data-source-name*;

Arguments

data-source-name

specifies the data source name that you created for the DBMS in the ODBC Administrator. DSN= is optional. If it is omitted, then the environment variable or default value is used. For the AS/400, the environment variable is AS400DSN, and the default value is **AS400**. For MS SQL Server, the environment variable is MSSQLDSN, and there is no default value. For more information about these environment variables, see or .

UID Statement

Specifies an alternate userid.

UID | **USER**=*user-id*;

Arguments

user-id

enables you to connect to the database with a user ID that is different from the default DBMS login connection ID. UID= is optional. When you use PROC CV2ODBC, you should use the same UID= and PW= values that were used to create the Version 6 view. PROC CV2ODBC validates the UID= and PASSWORD= values and uses them for the connection to the DBMS. If you do not know the UID or password that is specified in the Version 6 view, then you can specify any valid user ID and password for the connection to the DBMS provided you have the proper permissions for that DBMS.

PASSWORD Statement

Specifies the ODBC password.

PWD | PASSWORD | PW= *password*;

Arguments

password

specifies the DBMS password that is associated with your DBMS user ID. When you use PROC CV2ODBC, you should use the same UID= and PW= values that were used to create the Version 6 view. PROC CV2ODBC validates the UID= and PASSWORD= values and uses them for the connection to the DBMS. If you do not know the UID or password that is specified in the Version 6 view, then you can specify any valid user ID and password for the connection to the DBMS provided you have the proper permissions for that DBMS.

SAVE Statement

Saves the generated PROC SQL statements to a file.

SAVE <AS=>*filename*;

Arguments

filename

enables you to save the PROC SQL statements that are automatically generated by PROC CV2ODBC to an external file. You can modify this file as needed and submit it on another platform that supports ODBC. If you specify a value for SAVE=, then the PROC SQL statements are not automatically submitted.

SUBMIT Statement

Enables automatic submission of the PROC SQL statements that are generated by PROC CV2ODBC.

SUBMIT<*statements*>;

Arguments

statements

enables you to automatically submit the PROC SQL statements that are generated by PROC CV2ODBC. If you specify a value for SAVE=, then the statements are not automatically submitted for processing. The SUBMIT statement is not necessary if you have not specified a value for SAVE=.

REPLACE Statement

Enables replacement of existing views and files.

REPLACE ALL | VIEW | FILE;

Arguments

ALL

enables you to overwrite existing views and files of the same name. Replaces both an existing TO= view and a SAVE= file if they already exist.

FILE

replaces the SAVE= file if it already exists. If the file already exists, and if REPLACE FILE or REPLACE ALL are not specified, the generated PROC SQL code will be appended to the file.

VIEW

replaces the TO= view if it already exists

Examples

Example 1 - Converting an AS/400 View Descriptor

In this example, PROC CV2ODBC converts the Version 6 AS4ACC.INVOICE view descriptor to an ODBC view, AS4LIB.V8_INVOICE. The PROC SQL statements that are generated by PROC CV2ODBC are both automatically submitted and saved to an external file named AS4_INVOICE.SAS.

```
libname as4acc 'd:\lib\as4\acc';
libname as4lib 'd:\lib\as4\lib';

proc cv2odbc dbms=as400;
  from view=as4acc.invoice;
  to view=as4lib.V8_invoice;
  user='testuser';
  password='testpass';
  dsn='IBM AS/400 Database';
  save ='as4_invoice.sas';
  replace all;
  submit;
run;
```

PROC CV2ODBC generates the following PROC SQL statements.

```
PROC SQL DQUOTE=ANSI;
  CREATE VIEW AS4LIB.V8_INVOICE AS
  SELECT
    "INVOICENUM" AS INVOICE INFORMAT= 11.0
```

```

        FORMAT= 11.0 LABEL='INVOICENUM' ,
        "BILLEDTO" AS CUSTOMER INFORMAT= $8.
        FORMAT= $8. LABEL= 'BILLEDTO' ,
        "AMTBILLED" AS AMOUNT INFORMAT= DOLLAR20.2
        FORMAT= DOLLAR20.2 LABEL= 'AMTBILLED' ,
        "BILLEDDBY" AS BILLEDDBY INFORMAT= 11.0
        FORMAT= 11.0 LABEL= 'BILLEDDBY' ,
        "BILLEDON" AS BILLEDON INFORMAT= MMDDYY8.
        FORMAT= MMDDYY8. LABEL= 'BILLEDON' ,
        "PAIDON" AS PAIDON INFORMAT= DATE7.
        FORMAT= DATE7. LABEL= 'PAIDON'
FROM _CVLIB_."INVOICE"
(DBCONDITION = "WHERE PAIDON IS NOT NULL "
DBINDEX = NO
DBCOLUMNS = 'INVOICENUM, BILLEDTO,
AMTBILLED, BILLEDDBY, BILLEDON,
PAIDON'
SASDATEFMT = ("BILLEDON"= MMDDYY8.
"PAIDON"= DATE7.)
)
USING LIBNAME _CVLIB_ ODBC
AUTOCOMMIT = YES
NOPROMPT="DSN=IBM AS/400 Database;
UID=testuser;PWD=testpass;NAM=1;XDYNAMIC=0;
LIBVIEW=0;DBQ=SASDEMO"
;
QUIT;

```

Example 2 - Converting an MS SQL Server View Descriptor

In this example, PROC CV2ODBC converts the Version 6 MSQACC.ORDER view descriptor to an ODBC view, MSQLIB.ORDERS. The PROC SQL statements that are generated by PROC CV2ODBC are saved to an external file named MSSQL.SAS, but are not automatically submitted.

```

libname msqacc 'd:\lib\msq\acc';
libname msqlib 'd:\lib\msq\lib';

proc cv2odbc dbms=mssql;
  from view = msqacc.order (pw=testpass);
  to view = msqlib.orders;
  user = 'testuser';
  pass = 'testpass';
  dsn = 'Microsoft SQL Server Database';
  save = 'mssql.sas';
  replace all;
run;

```

PROC CV2ODBC generates the following PROC SQL statements.

```

PROC SQL DQUOTE=ANSI;
CREATE VIEW MSQLIB.ORDERS(PW=TESTPASS) AS
SELECT
  "ordernum" AS ORDERNUM INFORMAT= 11.0
  FORMAT= 11.0 LABEL= 'ordernum' ,
  "fabriccharges" AS AMOUNT INFORMAT= DOLLAR22.2

```

```

        FORMAT= DOLLAR22.2 LABEL= 'fabriccharges' ,
"shipto" AS SHIPTO INFORMAT= $8. FORMAT= $8.
        LABEL= 'shipto' ,
"dateordered" AS DATEORDE
        INFORMAT= DATETIME21.2
        FORMAT= DATETIME21.2 LABEL= 'dateordered' ,
"shipped" AS SHIPDATE INFORMAT= MMDDYY10.
        FORMAT= MMDDYY10. LABEL= 'shipped'
, "takenby" AS TAKENBY INFORMAT= 11.0
        FORMAT= 11.0 LABEL= 'takenby' ,
"processedby" AS PROCESSE INFORMAT= 11.0
        FORMAT= 11.0 LABEL= 'processedby'
FROM _CVLIB_."orders"
(DBCONDITION = "WHERE (takenby=456910 or
        takenby=234967 or
        dateordered='1988-11-03' or
        dateordered='12/23/1988')
        and shipped is not null"
SASDATEFMT = ("dateordered"=DATETIME21.2
        "shipped"=MMDDYY10.)
)
USING LIBNAME _CVLIB_ ODBC
PRESERVE_TAB_NAMES = YES
NOPROMPT="DSN=Microsoft SQL Server Database;
        SERVER=dbipcl.pc.sas.com;
        UID=testuser;PWD=testpass;DATABASE=sample";
QUIT;

```


Glossary

This glossary defines SAS System terms that are used in this book as well as terms that relate specifically to SAS/ACCESS software.

access descriptor

a SAS/ACCESS file that describes data to the SAS System that is in a DBMS table. You use an access descriptor as a master descriptor file from which to create view descriptors. See also view and view descriptor.

browsing data

the process of viewing the observations in a file. Depending on how the file is accessed, observations may be viewed one at a time or as a group in a tabular format. You cannot update data that you are browsing.

client

in a network, a workstation or application that requests services from a server. See also server.

column

a vertical component of a DBMS table. Each column has a unique name and contains data of a specific type and with certain attributes. A column is analogous to a variable in SAS terminology.

column function

an operation that is computed over each value in the column that is named as the argument of the function. For example, AVG(SALARY) is a column function.

data type

an attribute of every column in a table. The data type tells the operating system how much physical storage to set aside for the column as well as the type of data that the column contains. It is similar to the type attribute of SAS variables.

data value

in the SAS System, a unit of character or numeric information in a SAS data set. A data value represents one variable in an observation.

database

an organized collection of related data. In most relational DBMSs, a database contains objects such as tables, views, and indexes.

database management system (DBMS)

an integrated software package that enables you to create and manipulate data in the form of databases. See also relational database management system.

DATA step view

a type of SAS data set that consists of a stored DATA step program. Like other SAS data views, a DATA step view contains a definition of data that are stored elsewhere. You create a DATA step view by specifying the VIEW= option in a DATA statement.

The view's input data can come from one or more sources, including external files and other SAS data sets. Because a DATA step view only reads (opens for input) other files, you cannot update this view's underlying data. DATA step views can function only as input data sets.

editing data

the process of viewing a file with the ability to change its data. Depending on how the file is accessed, observations may be viewed one at a time or as a group in a tabular format.

engine

a part of the SAS System that reads from or writes to a file. Each engine allows the SAS System to access files that have a particular format. There are several types of engines. See also interface view engine.

file

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

format

an instruction that the SAS System uses to display or write each value of a variable. Some formats are supplied by SAS software. Other formats can be written by the user with the FORMAT procedure in base SAS software or created using SAS/TOOLKIT software. In SAS/ACCESS software, the default formats vary according to the interface product.

index

(1) in SAS software, a component of a SAS data set that enables the SAS System to access observations in the data set quickly and efficiently. The purpose of SAS indexes is to optimize WHERE-clause processing and to facilitate BY-group processing. (2) a named object that directs the database management system (DBMS) to the storage location of a particular data value for a given column. These indexes are also used to optimize WHERE-clause processing and join processing.

Depending on how selection criteria are specified, the SAS System can use DBMS indexes to speed data retrieval.

informat

a pattern that the SAS System uses to determine how values that are entered in variable fields should be interpreted. The SAS System provides a set of standard informats and also enables you to define your own custom informats.

interface view engine

a SAS System engine that retrieves data directly from files that have been formatted by other software vendors.

libref

the name that is temporarily associated with a SAS data library. You assign a libref with a LIBNAME statement or with operating system control language.

member

a SAS file in a SAS data library.

member name

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

member type

a name assigned by the SAS System that identifies the type of information stored in a SAS file. Member types include ACCESS, DATA, CATALOG, PROGRAM, and VIEW.

missing value

a value in the SAS System that indicates that no data are stored in the variable for the current observation. By default, the SAS System represents a missing numeric value with a single period, and a missing character value is represented by a blank space.

observation

the horizontal component of a SAS data file. An observation is a collection of data values that are associated with a single entity, such as a customer or state. Each observation contains one data value for each variable in the data file. An observation is analogous to a row in a DBMS table.

PROC SQL view

a SAS data set (of type VIEW) that is created by the SQL procedure. A PROC SQL view contains no data; instead, it stores the information that is required to read data values from its underlying files, which can include SAS data files, SAS/ACCESS views, DATA step views, or other PROC SQL views. When executed, a PROC SQL view's output can be either a subset or a superset of one or more underlying files. However, in the current release, you cannot reference a PROC SQL view to update its underlying data. See also view.

query

a request to retrieve data from a DBMS table or view. Queries are implemented with the PROC SQL SELECT statement.

referential integrity

consistency among related data that are stored in different tables.

relational database management system

a database management system that organizes and accesses data according to relationships between data items. ORACLE and DB2 are examples of relational database management systems.

row

the horizontal component of a DBMS table. A row is analogous to a SAS observation.

SAS data file

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

SAS data library

a collection of one or more SAS files that are recognized by the SAS System and that are referenced and stored as a unit. Each file is a member of the library.

SAS data set

(1) descriptor information and its related data values, organized as a table of observations and variables that can be processed by the SAS System. A SAS data set can be either a SAS data file or a SAS data view. (2) in comparison to database management systems, a collection of information that is stored as a unit under 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 compose an observation, and those in a column compose a variable. See also SAS data file and SAS data view.

SAS data view

one of the forms of a SAS data set that were implemented in Version 6 of the SAS System. A SAS data view contains only the descriptor and other information that is

required to retrieve the data values from other data sources. PROC SQL views, SAS/ACCESS views, and DATA step views are considered SAS data views. SAS data views are of member type view.

SAS Explorer

an interactive, windowing interface to SAS System software. Display manager commands can be issued by typing them on the command line, pressing function keys, or selecting items from the menu facility. Within one session, many different tasks can be accomplished, including preparing and submitting programs, viewing and printing results, and debugging and resubmitting programs.

SAS/ACCESS views

See view descriptor and SAS data view.

server

in a network, a computer or application that is reserved for servicing other computers or applications in the network. Servers can provide file services and communication services, and they enable users to access common resources such as disks, data, and modems. See also client.

SQL Procedure Pass-Through facility

a group of SQL procedure statements that send and receive data directly between a relational database management system (DBMS) and the SAS System. The Pass-Through facility includes the CONNECT, DISCONNECT, and EXECUTE statements, and the CONNECTION TO component. A SAS/ACCESS interface view engine is used to connect the two systems; therefore, you must have SAS/ACCESS software installed for the DBMS that you use in order to use the Pass-Through Facility.

Structured Query Language (SQL)

the standardized, high-level query language that is used in relational database management systems to create and manipulate database management system objects. The SAS System implements SQL through the SQL procedure.

trigger

for some DBMSs, a type of user-defined stored procedure that is executed whenever a user issues a data-modification command such as INSERT, DELETE, or UPDATE for a specified table or column. Triggers can be used to implement referential integrity or to maintain business constraints. See also referential integrity.

variable

a column in a SAS data set. A variable is a set of data values that describe a given characteristic across all observations. In the ACCESS procedure, variables are created from the DBMS table's columns or fields.

view

a definition of a virtual data set. The definition is named and stored for later use. This file contains no data but describes or defines data that are stored elsewhere. See also DATA step view, PROC SQL view, SAS data view, and view descriptor.

view descriptor

a SAS/ACCESS file that defines all or a subset of the DBMS data that are described by an access descriptor. See also access descriptor.

wildcard character

a character that is used to represent possible characters at a particular position in a word in order to generalize the word.

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

SAS/ACCESS® Software for Relational Databases: Reference, Version 8

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

ISBN 1-58025-558-2

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.