



THE
POWER
TO KNOW.

**SAS[®] Enterprise Miner[™] and
SAS[®] Text Miner Procedures
Reference for SAS[®] 9.1.3
The DMDB Procedure
(Book Excerpt)**

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., SAS® *Enterprise Miner™* and SAS® *Text Miner Procedures: Reference for SAS® 9.1.3*, Cary, NC: SAS Institute Inc.

SAS® Enterprise Miner™ and SAS® Text Miner Procedures: Reference for SAS 9.1.3

Copyright © 2008 by SAS Institute Inc., Cary, NC, USA.

All rights reserved. Produced in the United States of America.

For a Web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of this 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).

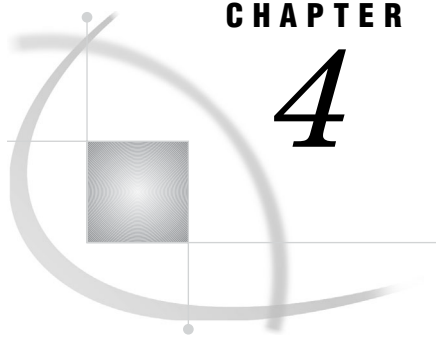
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, October 2008

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at support.sas.com/publishing or call 1-800-727-3228.

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.



CHAPTER

4

The DMDB Procedure

<i>Overview: DMDB Procedure</i>	53
<i>Syntax: DMDB Procedure</i>	53
<i>PROC DMDB Statement</i>	54
<i>CLASS Statement</i>	56
<i>FREQ Statement</i>	57
<i>ID Statement</i>	57
<i>TARGET Statement</i>	58
<i>VAR Statement</i>	58
<i>Details: DMDB Procedure</i>	58
<i>Examples: DMDB Procedure</i>	59
<i>Example 1: Getting Started with the DMDB Procedure</i>	59
<i>Example 2: Specifying a FREQ Variable</i>	61
<i>References</i>	62

Overview: DMDB Procedure

SAS Enterprise Miner architecture is based on the creation of a data mining database (DMDB) that is a snapshot of the original data. PROC DMDB creates this DMDB from the input data source. It also compiles and computes metadata information about the input data based on variable roles and stores it in a metadata catalog. The DMDB and the associated metadata catalog facilitate subsequent data mining activities. They are both designed for processing and storage efficiencies.

Note: The DMDBCAT= argument is required for the following procedures: ASSOC, DMDB, DMINE, DMNEURAL, DMSPLIT, DMREG, MBR, NEURAL, DMVQ, SEQ, and SVM. It is optional in the SPLIT procedure. It is not a valid argument in the ARBOR, RULEGEN, STDIZE, and TAXONOMY procedures. △

Syntax: DMDB Procedure

```

PROC DMDB <option(s)>;
    CLASS variable(s) <order-option(s)>;
    FREQ variable;
    ID variable(s);
    TARGET variable(s);
    VAR variable(s) ;
  
```

PROC DMDB Statement

Invokes the DMDB procedure.

```
PROC DMDB <option(s)>;
```

Required Arguments

DATA=<libref.> SAS-data-set

Names the SAS data set containing the information that you want added to the data mining database.

Options

BATCH | NOMETA

Specifies the creation of a new metadata catalog as specified in the DMDBCAT= option, instead of updating an existing metadata catalog. Any existing catalog with this name will be overridden and replaced by system-generated information.

CLASSOUT=<libref.> SAS-data-set

Specifies that metadata for the CLASS variables be output to this data set. The output data set contains these columns:

NAME

The name of the included class variable.

LEVEL

The levels of the class variables included in the data set.

FREQUENCY

The total number of times each level of each variable is included in the data set.

TYPE

Tells whether the levels of the variables are character strings or numeric.

CRAW

If the levels are character strings then this displays the raw input form of the levels. If the levels are numeric then it is shown as missing.

NRAW

If the levels are numeric then this displays the raw input form of the levels. If the levels are character strings then a missing value is given.

FREQPERCENT

This gives the observed percent of each level within a particular variable by including missing values.

NMISSPERCENT

This updates the percent of each level within a variable after removing missing values.

MISSINGONLY

Used to obtain frequencies and other applicable statistics for CLASS and VAR variables based only on distinction between missing and nonmissing values.

MAXLEVEL=*integer*

Specifies the maximum number of class levels to be processed.

Default: MACINT. (If an integer greater than MACINT is specified, the integer specified for MAXLEVEL is ignored and MACINT is used.)

Range: *integer* ≥ 3

NONORM

Specifies that class level values be generated without normalization. Normalization involves left justification, truncation to NORMLEN length, and upcasing. By default PROC DMDB generates metadata with normalized class levels. Normalization helps with unclean or inconsistent data. For example for a class variable RESPONSE, values of "Yes", "YES", " yes" will all map into a single normalized level "YES". At the same time, also note that class levels need to be unique within the default normlen of 32 characters. For example, for a variable ITEMDESC, values such as "ABC Department store, nylon socks blue" and "ABC Department store, nylon socks black" will map into a single normalized level of " ABC DEPARTMENT STORE, NYLON SOCKS ".

NORMLEN=*integer*

Specifies the normalized length of class level values (category values). Normalization involves left justification, truncation to NORMLEN length, and upcasing. See NONORM option.

Default: 32

OUT=*<libref.> SAS-data-set*

Names the data mining database (DMDB) that you want created. The new DMDB contains each of the ID, VAR, and FREQ variables – copied 'as is' from the DATA= data set. The DMDB also contains each of the CLASS variables written out as their corresponding integer class level value in 5–bytes (sizeof(float)+1).

PMML

Specifies that PMML score code be generated. PROC DMDB produces the DataDictionary component of a PMML document. For more information, see the PMML Support in Enterprise Miner section in the Enterprise Miner 5.3 Java Help.

VARDEF=*divisor*

Specifies the divisor to use in the calculation of the variance and standard deviation. The following table shows the possible values for VARDEF:

Table 4.1 Values for VARDEF=

Value	Divisor	Formula
DF	degrees of freedom	$n - 1$
N	number of observations	n

Default: DF (degrees of freedom)

VAROUT=*<libref.> SAS-data-set*

Specifies that metadata for the VAR variables be output to this data set. This output data set contains these columns:

NAME

The name of the included variable.

NMISS

The number of missing values for each variable.

N

The total number of observations of a particular variable, not including missing values.

MIN

The smallest observed value of an included variable.

MAX

The largest observed value of an included variable.

MEAN

The average of values of an included variable, not including missing values.

STD

The standard deviation of an included variable, not including missing values.

SKEWNESS

The skewness of an included variable, not including missing values.

KURTOSIS

The kurtosis of an included variable, not including missing values.

CLASS Statement

Specifies the variables whose values define subgroup combinations for the analysis.

INTERACTION: CLASS, ID, and VAR statements are mutually exclusive.

CLASS *variable(s)* <*order-option(s)*>;

Required Argument

variable(s)

Specifies one or more categorical variables to be used in the analysis. For each CLASS variable, the metadata contains information on each of the following: its class level value, its frequency, and its ordering information.

Range: *variable(s)* can be character or numeric.

Options

ORDER

Specifies the order to use when considering the levels of classification variables (specified in the CLASS statement) to be sorted. *order-option(s)* can be one of the following:

ASCENDING | ASC

Class levels are arranged in lowest-to-highest order of unformatted values.

DESCENDING | DESC

Class levels are arranged in highest-to-lowest order of unformatted values.

ASCFORMATTED | ASCFMT

Class levels are arranged in ascending order by their formatted values.

DESFORMATTED | DESFMT

Class levels are arranged in descending order by their formatted values.

DSORDER | DATA

Class levels are arranged according to the order of their appearance in the input data set.

Default: ASCENDING

FREQ Statement

Specifies a numeric variable that contains the frequency of each observation.

Tip: Using the FREQ variable in PROC DMDB ensures that the FREQ variable is automatically used by all other Enterprise Miner procedures in the project.

FREQ *variable*;

Required Argument

variable

Specifies a numeric variable whose value represents the frequency of the observation.

Default: If *variable* is 0 or missing, then the observation is omitted in the DMDB and is not included in statistical calculations.

ID Statement

Includes additional variables in the output data set.

INTERACTION: CLASS, ID, and VAR statements are mutually exclusive.

ID *variable(s)*;

Required Argument

variable(s)

Identifies one or more variables from the input data set whose maximum values for groups of observations are included in the output data set by PROC DMDB.

Range: *variable(s)* can be character or numeric.

TARGET Statement

Specifies variables to be created for the output data set.

TARGET *variable(s)*;

Required Argument

variable(s)

Identifies TARGET variables. Variables must also be specified in the VAR or CLASS variable lists.

Range: *variable(s)* can be character or numeric.

VAR Statement

Identifies the analysis variables and their order in the results.

INTERACTION: CLASS, ID, and VAR statements are mutually exclusive.

Alias: VAR

VAR *variable(s)* ;

Required Argument

variable(s)

Identifies the analysis variables and specifies their order in the results.

The metadata catalog contains the following statistics for the VAR variables: N, NMISS, MIN, MAX, SUM, CSS, USS, STD, SKEWNESS, and KURTOSIS.

Default: If you omit the VAR statement, PROC DMDB analyzes all numeric variables not listed in the other statements.

Range: *variable(s)* are numeric only.

Details: DMDB Procedure

The data mining database (DMDB) is maintained as a SAS data set. The metadata information associated with the DMDB is maintained in a SAS catalog. Metadata includes overall data set information as well as statistical information for the variables according to their roles. For each CLASS variable, the metadata contains information on each of the following: its class level value, its frequency, and its ordering information. In the DMDB, the CLASS variables are stored as integers 0, 1, 2, ..., which can be mapped into different class level values.

For each VAR variable, the metadata catalog contains the following statistics:

N	The number of observations with nonmissing values of the variable
NMISS	The number of observations with missing values of the variable
MIN	The minimum
MAX	The maximum
SUM	The sum of all the nonmissing values of the variable
CSS	The corrected sum of squares
USS	The uncorrected sum of squares
STD	The standard deviation
SKEWNESS	Measure of the tendency for the distribution of values to be more spread out on one side of the mean than on the other
KURTOSIS	Measure of the “heaviness of the tails”

(Refer to the *SAS Procedures Guide*, Chapter 1 for formulas and other details.)

DMDBs are created only for training data and should not be used for validation or test during modeling.

Examples: DMDB Procedure

The following examples were executed using the Windows XP operating system and the SAS software release 9.1.3 Service Pack 4.

Example 1: Getting Started with the DMDB Procedure

Features:

- Specifying the Output DMDB Data Set and Catalog
 - Defining the Numeric Variables in a VAR Statement
 - Defining the Class Variables in a Class Statement
 - Setting the Order of the Class Variables
 - Defining the Target Variable in a Target Statement
-

This example demonstrates how to create a data mining database (DMDB) data set and catalog. The example uses the fictitious mortgage data set name `SAMPSIO.HMEQ`. The data set contains 5,960 cases. Each case represents an applicant for a home equity

loan. All applicants have an existing mortgage. The binary target BAD indicates whether an applicant eventually defaulted or was ever seriously delinquent. There are ten numeric inputs and two class inputs available for subsequent modeling.

Program

The PROC DMDB statement invokes the procedure. The BATCH option requests the creation of a new DMDB catalog. The DATA= option specifies the input data set.

```
proc dmdb batch data=sampsio.hmeq
```

The OUT= option specifies the name of the output DMDB data set. The DMDBCAT= option specifies the name of the output DMDB catalog.

```
    out=dmhmeq
    dmdbcat=cathmeq;
```

The VAR statement identifies the numeric analysis variables. If you omit the VAR statement, PROC DMDB analyzes all numeric variables not listed in other statements.

```
    var loan derog mortdue value yoj delinq
        clage ninq clno debtinc;
```

The CLASS statement specifies the categorical variables to be used in the analysis. The ORDER option specifies the order to use when considering the levels of the classification variables. Valid ORDER options include ASCENDING (ASC), DESCENDING (DESC), ASCFORMATTED (ASCFMT), DESFORMATTED (DEFMT), or DSORDER (DATA). The default for the ORDER option is set to ASCENDING.

```
    class bad(desc)
        reason(ascending)
        job;
```

The TARGET statement identifies the target (response) variable.

```
    target bad;
run;
```

Log

```
1  proc dmdb batch data=sampsio.hmeq
2
3      out=dmhmeq
4      dmdbcat=cathmeq;
5
6      var loan derog mortdue value yoj delinq
7          clage ninq clno debtinc;
8
9      class bad(desc)
10         reason(ascending)
```

```

11          job;
12
13          target bad;
14  run;

```

```

NOTE: Records processed = 5960   Memory used = 511K.
NOTE: There were 5960 observations read from the data set SAMPSIO.HMEQ.
NOTE: The data set WORK.DMHMEQ has 5960 observations and 13 variables.
NOTE: PROCEDURE DMDB used (Total process time):
      real time           0.10 seconds
      cpu time            0.06 seconds

```

Example 2: Specifying a FREQ Variable

Features Specifying a FREQ variable with the FREQ Statement

This example demonstrates how to define a FREQ variable in the DMDB data set and catalog. A FREQ variable represents the frequency of occurrence for other values in each observation of the input data set. The DATA step required to create the WORK.FREQEX input data set is provided.

Program

```

data freqex;
  input count x1 x2 x3 Y ;
  datalines;
3      -0.17339      -0.04926      -0.61599      0
2      -1.51586      0.31526      -1.65430      1
1      1.04348      0.64517      -0.06878      0
1      -1.74298      0.02592      -0.71203      1
1      0.07806      1.45284      -0.39064      1
4      0.20073      0.22533      -0.44507      0
1      -0.08887      -1.24641      -0.73156      0
1      0.10309      0.88542      -1.63595      1
2      -0.57030      -1.35613      -1.58209      0
1      -1.39170      -1.22333      1.98124      1
2      0.51356      -0.36128      0.77962      0
1      -0.89216      -0.01054      -0.76720      0
1      -0.09882      1.43263      0.53820      0
3      0.03225      -0.17737      0.25381      0
1      -0.14203      -1.64183      -0.34028      0
1      -0.24436      -0.83537      -2.00245      0
2      -0.78277      0.00284      -0.75016      0
1      0.77732      -0.28847      -0.77437      0
1      1.55172      -0.21167      -0.53833      0
2      -0.74054      -1.23276      0.11452      1
run;

proc dmdb batch data=freqex out=dmfout dmdbcatalog=outfcatalog;
  var x1 x2 x3;

```

```
class y(desc);
target y;
```

The **FREQ** statement specifies the numeric variable that contains the frequency of each observation.

```
freq count;
run;
```

Log

```
1 data freqex;
2 input count X1 X2 X3 Y ;
3 datalines;
NOTE: The data set WORK.FREQEX has 20 observations and 5 variables.
NOTE: DATA statement used (Total process time):
real time 0.01 seconds
cpu time 0.01 seconds
24 run;
25
26 proc dmdb batch data=freqex out=dmfout dmdbcat=outfcats;
27 var x1 x2 x3;
28 class y(desc);
29 target y;
30
31 freq count;
32 run;
NOTE: Records processed = 20 Memory used = 511K.
NOTE: There were 20 observations read from the data set WORK.FREQEX.
NOTE: The data set WORK.DMFOUT has 20 observations and 5 variables.
NOTE: PROCEDURE DMDB used (Total process time):
real time 0.06 seconds
cpu time 0.01 seconds
```

References

SAS Institute Inc. (2008), *PMML Support in Enterprise Miner*, in SAS Enterprise Miner 5.3 Java Help.