CHAPTER

12

Formats

Formats in the OS/390 Environment

In general, formats are completely portable. Only the formats that have aspects specific to OS/390 are documented in this chapter. All portable formats are described in *SAS Language Reference: Dictionary*; that information is not repeated here. Instead, you are given details on how the format behaves in the OS/390 environment, then you are referred to *SAS Language Reference: Dictionary* for further details.

Considerations for Using Formats in the OS/390 Environment

EBCDIC and Character Data

The following character formats produce different results on different computing platforms, depending on which character-encoding system the platform uses. Because OS/390 uses the EBCDIC character-encoding system, all of the following formats convert data from EBCDIC.

These formats are not discussed in detail in this chapter because the EBCDIC character-encoding system is their only host-specific aspect.

$ASCII*w.*
converts EBCDIC character data to ASCII character data.

$BINARY*w.*
converts EBCDIC character data to binary representation, where each character is represented by eight binary characters.

$EBCDIC*w.*
converts EBCDIC data to character data. Under OS/390, $EBCDIC*w.* and $CHAR*w.* are equivalent.

$HEX*w.*
converts EBCDIC character data to hexadecimal representation.

$OCTAL*w.*
converts EBCDIC character data to octal representation.

All the information that you need in order to use these formats under OS/390 is in *SAS Language Reference: Dictionary*.

Floating-Point Number Format and Portability

The manner in which OS/390 stores floating-point numbers can affect your data. See "Representation of Floating-Point Numbers" on page 143 for details.

Writing Binary Data

If a SAS program that writes binary data is run in only one operating environment, you can use the following native-mode formats:*

IB*w.d*
writes integer binary (fixed-point) values, including negative values, that are represented in two's complement notation.

PD*w.d*
writes data that are stored in IBM packed decimal format.

PIB*w.d*
writes positive integer binary (fixed-point) values.

RB*w.d*
writes real binary (floating-point) data.

If you want to write SAS programs that can be run on multiple machines that use different byte-storage systems, use the following IBM 370 formats:

S370FF*w.d*
is used on other computer systems to read EBCDIC data from IBM mainframe files.

S370FIB*w.d*
writes integer binary data in IBM mainframe format.

S370FIBU*w.d*
writes unsigned integer binary data in IBM mainframe format.

S370FPD*w.d*
writes packed decimal data in IBM mainframe format.

S370FPDU*w.d*
writes unsigned packed decimal data in IBM mainframe format.

* Native-mode formats use the byte-ordering system that is standard for the operating environment.

S370FPIB*w.d*
writes positive integer binary data in IBM mainframe format.

S370FRB*w.d*
writes real binary data in IBM mainframe format.

S370FZD*w.d*
writes zoned decimal data in IBM mainframe format.

S370FZDL*w.d*
writes zoned decimal leading sign data in IBM mainframe format.

S370FZDS*w.d*
writes zoned decimal separate leading sign data in IBM mainframe format.

S370FZDT*w.d*
writes zoned decimal separate trailing sign data in IBM mainframe format.

S370FZDU*w.d*
writes unsigned zoned decimal data in IBM mainframe format.

These IBM 370 formats enable you to write SAS programs that can be run in any SAS environment, regardless of the standard for storing numeric data. They also enhance your ability to port raw data between host operating environments.

For more information about the IBM 370 formats, see *SAS Language Reference: Dictionary*.

BEST*w.*

SAS System chooses the best notation

Numeric

Width range: 1-32 bytes

Default width: 12

Alignment: right

OS/390 specifics: writes output as EBCDIC, minimum and maximum values

Details

Numbers are written using EBCDIC code with one digit per byte. Because the value is output as EBCDIC text characters, you can print it without further formatting.

The range of the magnitude of numbers is from 5.4×10^{-79} to 7.2×10^{75}. Any number that is outside this range causes an overflow error. All numeric variables that are represented by the SAS System are within this range.

The following examples illustrate the use of BEST*w.* under OS/390:

Value	Format	Results	Notes
1234	best6.	**bb1234**	
-1234	best6.	**b-1234**	
12.34	best6.	**b12.34**	
12345678	best8.	**1.2346E8**	truncated and rounded

Note: In these examples, the Value column represents the value of the SAS numeric variable. The Results column shows what the numeric output looks like when viewed from a text editor. The b characters in the Results column indicate blank spaces. See Table 9.2 on page 146 for a table of commonly used EBCDIC characters. △

See Also

□ Format: BEST*w.* in *SAS Language Reference: Dictionary*

Ew.

Writes numeric values in scientific notation

Numeric

Width range: 7- 32 bytes

Default width: 12

Alignment: right

OS/390 specifics: writes output as EBCDIC, minimum and maximum values

Details

Numbers are represented using the EBCDIC code, with one digit per byte. Because the values are stored in EBCDIC, they can be printed without further formatting.

The range of the magnitude of numbers is from 5.4×10^{-79} to 7.2×10^{75}. Any number that is outside of this range causes an overflow error. All numeric variables that are represented by the SAS System are within this range.

The following examples illustrate the use of E*w.* under OS/390:

Value	Format	Results	Notes
123	e10.	**b1.230E+02**	
-123	e10.	**-1.230E+02**	
12.3	e10.	**b1.230E+01**	
12345678	e10.	**b1.235E+07**	truncated and rounded

Note: In these examples, the Value column represents the value of the SAS numeric variable. The Results column shows what the numeric value looks like when viewed from a text editor. The b characters in the Results column indicate blank spaces. See Table 9.2 on page 146 for a table of commonly used EBCDIC characters. △

See Also

- Format: E*w.* in *SAS Language Reference: Dictionary*
- Informat: "E*w.d*" on page 208

HEX*w.*

Converts real binary (floating-point) values to hexadecimal representation

Numeric

Width range: 1-16 bytes

Default width: 8

Alignment: left

OS/390 specifics: writes output as EBCDIC, IBM floating-point format

Details

Each hexadecimal digit is written using the EBCDIC code, which requires one byte per digit. See Table 9.2 on page 146 for a table of commonly used EBCDIC characters.

The format of floating-point numbers is host-specific. See "Representation of Floating-Point Numbers" on page 143 for a description of the IBM floating-point format that is used under OS/390.

The *w* value of the HEX*w.* format determines whether the number is written as a floating-point number or as an integer. When you specify a width value of 1 through 15, the real binary numbers are truncated to fixed-point integers before being converted to hexadecimal representation. When you specify 16 for the width, the floating point values are used, and the numbers are not truncated.

The following examples illustrate the use of HEX*w.* under OS/390:

Value	Format	Results	Notes
31.5	hex16.	**421F800000000000**	floating-point number
31.5	hex15.	**00000000000001F**	integer
-31.5	hex16.	**C21F800000000000**	floating-point number
-31.5	hex15.	**FFFFFFFFFFFFFE1**	integer

Note: In these examples, the Value column represents the value of the SAS numeric variable. The Results column shows what the numeric value looks like when viewed from a text editor. △

See Also

- Format: HEX*w.* in *SAS Language Reference: Dictionary*
- Informat: "HEX*w.*" on page 208
- "Representation of Numeric Variables" on page 143

IB*w.d*

Writes numbers in integer binary (fixed-point) format

Numeric

Width range: 1-8 bytes

Default width: 4

Decimal range: 0-10

Alignment:

OS/390 specifics: two's complement notation

Details

On an IBM mainframe system, integer values are stored in two's complement notation.

If an overflow occurs, the value written is the largest value that fits into the output field; the value will be positive, negative, or unsigned, as appropriate. If the format includes a *d* value, the number is multiplied by 10^d.

Here are some examples of the IB*w.d* format:

Value	Format	Results (Hex)	Notes
-1234	ib4.	**FFFFFB2E**	
12.34	ib4.	**0000000C**	
123456789	ib4.	**075BCD15**	
1234	ib6.2	**00000001E208**	a *d* value of 2 causes the number to be multipled by 10^2
-1234	ib6.2	**FFFFFFFE1DF8**	a *d* value of 2 causes the number to be multipled by 10^2
1234	ib1.	**7F**	overflow occurred
-1234	ib1.	**80**	overflow occurred

Note: In these examples, the Value column represents the value of the numeric variable. The Results column shows a hexadecimal representation of the bit pattern written by the corresponding format. (You cannot view this data in a text editor, unless you can view it in hexadecimal representation.) △

See Also

- Formats: IB*w.d*, S370FIB*w.d*, and S370FPIB*w.d* in *SAS Language Reference: Dictionary*
- Informat: “IB*w.d* ”on page 209

PD*w.d*

Writes values in IBM packed decimal format

Numeric

Width range: 1-16 bytes

Default width: 1

Decimal range: 0-31

Alignment: left

OS/390 specifics: IBM packed decimal format

Details

In packed decimal format, each byte represents two decimal digits. An IBM packed decimal number consists of a sign and up to 31 digits, thus giving a range of 10^{31} –1 to -10^{31} + 1. The sign is written in the rightmost nibble. (A nibble is four bits or half a byte.) A hexadecimal C indicates a plus sign, and a hexadecimal D indicates a minus sign. The rest of the nibbles to the left of the sign nibble represent decimal digits. The hexadecimal values of these digit nibbles correspond to decimal values; therefore, only values between ’0’x and ’9’x can be used in the digit p sitions.

If an overflow occurs, the value that is written is the largest value that fits into the output field; the value will be positive, negative, or unsigned, as appropriate.

Here are several examples of packed decimal format:

Value	Format	Results (Hex)	Notes
-1234	pd3.	**01234D**	
1234	pd2.	**999C**	overflow occurred
1234	pd4.	**0001234C**	
1234	pd4.2	**0123400C**	a *d* value of 2 causes the number to be multiplied by 10^2

Note: In these examples, the Value column represents the value of the data, and the Results column shows a hexadecimal representation of the bit pattern written by the corresponding format. (You cannot view this data in a text editor, unless you can view it in hexadecimal representation.) △

See Also

- Formats: PD*w.d* and S370FPD*w.d* in *SAS Language Reference: Dictionary*
- Informat: "PD*w.d*" on page 210

RB*w.d*

Writes numeric data in real binary (floating-point) notation

Numeric
Width range: 2-8 bytes
Default width: 4
Decimal range: 0-10
Alignment: left
OS/390 specifics: IBM floating-point format

Details

The format of floating-point numbers is host-specific. See "Representation of Floating-Point Numbers" on page 143 for a description of the format that is used to store floating-point numbers under OS/390.

Here are some examples of how decimal numbers are written as floating-point numbers using the RB*w.d* format:

Value	Format	Results (Hex)	Notes
123	rb8.1	**434CE00000000000**	
123	rb8.2	**44300C0000000000**	
-123	rb8.	**C27B000000000000**	
1234	rb8.	**434D200000000000**	
1234	rb2.	**434D**	truncation occurred
12.25	rb8.	**41C4000000000000**	

Note: In these examples, the Value column represents the value of the data, and the Results column shows a hexadecimal representation of the bit pattern written by the corresponding format. (You cannot view this data in a text editor, unless you can view it in hexadecimal representation.) △

See Also

- Formats: RB*w.d* and S370FRB*w.d* in *SAS Language Reference: Dictionary*
- Informat: "RB*w.d* "on page 211

w.d

Writes numeric data

Numeric

Width range: 1-32 bytes

Default width: 12

Decimal range: *d*<*w*

Alignment: right

OS/390 specifics: writes output as EBCDIC, minimum and maximum values

Details

The *w.d* format writes numeric values one digit per byte using EBCDIC code. Because the values are stored in EBCDIC, they can be printed without further formatting.

Numbers written with the *w.d* format are rounded to the nearest number that can be represented in the output field. If the number is too large to fit, the BEST*w.d* format is used. Under OS/390, the range of the magnitude of numbers that can be written with the BEST*w.d* format is from 5.4×10^{-79} to 7.2×10^{75}.

The following examples illustrate the use of the *w.d* format:

Value	Format	Results
1234	4.	`1234`
1234	5.	`b1234`
12345	4.	`12E3`
123.4	6.2	`123.40`
-1234	6.	`b-1234`

Note: In these examples, the Value column represents the value of the data, and the Results column shows what the numeric value looks like when viewed from a text editor. The b characters in the Results column indicate blank spaces. See Table 9.2 on page 146 for a table of commonly used EBCDIC characters. △

See Also

□ Format: *w.d* in *SAS Language Reference: Dictionary*

ZD*w.d*

Writes zoned decimal data

Numeric

Width range: 1-32 bytes

Default width: 1

Decimal range: 0-32

Alignment: left

OS/390 specifics: IBM zoned decimal format

Details

Like standard format, zoned decimal digits are represented as EBCDIC characters. Each digit requires one byte. The rightmost byte represents both the least significant digit and the sign of the number. Digits to the left of the least significant digit are written as the EBCDIC characters 0 through 9. The character that is written for the least significant digit depends on the sign of the number. Negative numbers are represented as the EBCDIC printable hexadecimal characters D0 through D9 in the least significant digit position, and positive numbers are represented as hexadecimal C0 through C9.

If an overflow occurs, the value that is written is the largest value that fits into the output field; the value will be positive, negative, or unsigned, as appropriate.

The following examples illustrate the use of the zoned decimal format:

Value	Format	Results (Hex)	Notes
1234	zd8.	**F0F0F0F0F1F2F3C4**	
123	zd8.1	**F0F0F0F0F1F2F3C0**	
123	zd8.2	**F0F0F0F1F2F3F0C0**	
-123	zd8.	**F0F0F0F0F0F1F2D3**	
0.000123	zd8.6	**F0F0F0F0F0F1F2C3**	
0.00123	zd8.6	**F0F0F0F0F1F2F3C0**	
1E-6	zd8.6	**F0F0F0F0F0F0F0C1**	overflow occurred

Note: In these examples, the Value column represents the value of the data, and the Results column shows a hexadecimal representation of the bit pattern that is written by the corresponding format. (You cannot view this data in a text editor unless you view it in hexadecimal representation.) See Table 9.2 on page 146 for a table of commonly used EBCDIC characters. △

See Also

- Formats: ZD*w.d*, S370FZDL*w.d*, S370FZDS*w.d*, S370FZDT*w.d*, and S370FZDU*w.d* in *SAS Language Reference: Dictionary*
- Informats: "ZD*w.d* "on page 212, "ZDB*w.d*" on page 213, and "S370FZD*w.d*" in *SAS Language Reference: Dictionary*

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Companion for the OS/390 Environment, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS® Companion for the OS/390® Environment, Verison 8

1–58025–523–X

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

1st printing, November 1999