CHAPTER
11

Formats

Formats Under UNIX

This chapter describes SAS formats that have behavior or syntax that is specific to UNIX environments. Each format description includes a brief "UNIX specifics" section that tells which aspect of the data set option is specific to UNIX. Each format is described in both this documentation and in *SAS Language Reference: Dictionary*.

HEX*w.* format

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

Numeric

Width range: 1 to 16

Default width: 8

Alignment: left

UNIX specifics: floating-point representation

Details

The HEX*w.* format converts a real (floating-point) binary number to its hexadecimal representation. When you specify a width value of 1 through 15, the real binary number is truncated to a fixed-point integer before being converted to hex. When you specify 16 for the width, the SAS System writes the floating-point value of the number but does not truncate it.

Note: UNIX systems vary widely in their floating-point representation. See "Reading and Writing Binary Data" on page 159 for more information. △

$HEX*w.* format

Converts character values to hexadecimal representation

Character
Width range: 1 to 32767
Default width: 4
Alignment: left
UNIX specifics: produces ASCII codes

Details

Under UNIX, the $HEX*w.* format produces hexadecimal representations of ASCII codes for characters, with each byte requiring two columns. Therefore, you need twice as many columns to output a value with the $HEX*w.* format.

IB*w.d* format

Writes integer binary values

Numeric
Width range: 1 to 8
Default width: 4
Decimal Range: 0–10
Alignment: left
UNIX specifics: byte order

Details

The IB*w.d* format writes integer binary (fixed-point) values. Integers are stored in integer-binary, or fixed-point, form. For example, the number 2 is stored as 00000002. If the format includes a *d* value, the data value is multiplied by 10^d.

For more details, see "Reading and Writing Binary Data" on page 159 .

PD*w.d* format

Writes packed decimal data

Numeric
Width range: 1 to 16
Default width: 1
Decimal Range: 0–31
Alignment: left
UNIX specifics: data representation

Details

The PD*w.d* format writes values in packed decimal format. In packed decimal data, each byte contains two digits. The *w* value represents the number of bytes, not the

number of digits. The value's sign is the first byte. Because the entire first byte is used for the sign, you should specify at least a width of 2.

For more details, see "Reading and Writing Binary Data" on page 159 .

PIB*w.d* format

Writes positive integer binary values

Numeric

Width range: 1 to 8

Default width: 1

Decimal Range: 0–10

Alignment: left

UNIX specifics: byte order

Details

The PIB*w.d* format writes fixed-point binary values, treating all values as positive. Thus, the high-order bit is part of the value, rather than the value's sign. If a *d* value is specified, the data value is multiplied by 10^d.

For more details, see "Reading and Writing Binary Data" on page 159 .

RB*w.d* format

Writes real binary (floating-point) data

Numeric

Width range: 2 to 8

Default width: 4

Decimal Range: 0–10

Alignment: left

UNIX specifics: floating-point representation

Details

The RB*w.d* format writes numeric data in real binary (floating-point) notation. The SAS System stores all numeric values in floating-point.

Real binary is the most efficient format for representing numeric values because the SAS System already represents numbers this way and no conversion is needed.

For more details, see "RB*w.d* informat" on page 209 and "Reading and Writing Binary Data" on page 159 .

ZD*w.d* format

Writes zoned decimal data

Numeric

Width range: 1 to 32

Default width: 1

Alignment: left

UNIX specifics: data representation

Details

The ZD*w.d* format writes zoned decimal data. This format is also known as overprint trailing numeric format. Under UNIX, the last byte of the field includes the sign along with the last digit. The conversion table for the last byte is as follows:

Digit	ASCII Character	Digit	ASCII Character
0	{	-0	}
1	A	-1	J
2	B	-2	K
3	C	-3	L
4	D	-4	M
5	E	-5	N
6	F	-6	O
7	G	-7	P
8	H	-8	Q
9	I	-9	R

For more details, see "ZD*w.d* informat" on page 211 and "Reading and Writing Binary Data" on page 159 .