

Table Display Precision

The display of data in a column of a table depends on the data type of the displayed wave, the values stored in the wave, and the numeric format settings applied to the column. In some cases, the cell area displays less than full precision but the entry line always displays full precision.

For example, this table displays five double-precision floating point waves using different display formats. Each of the waves contains the values 10, 3.141592653589793, and 3 billion.

Table1:General,Integer,Decimal,Hex,Octal					
Point	General	Integer	Decimal	Hex	Octal
0	10	10	10.000	0000000A	00000000012
1	3.14159	3	3.142	# Can't format	# Can't format
2	3e+09	3000000000	3000000000.0	B2D05E00	26264057000
3					

The first column is set to general format with six digits of precision which is the factory default format for new table columns. The general format chooses either integer, floating point or scientific notation depending on the displayed value.

The value of General[1] is displayed in the body of the table as 3.14159 since the format specifies six digits of precision. Since this cell is selected as the target cell, its value appears in the entry line. However, the entry line displays the value using full 16-digit precision rather than using the column's format. This guarantees that, if you click in the entry line and accept the text that is already there, no precision is lost.

If the General wave were single-precision floating point instead of double-precision, the entry line would show 8 digits for non-integer values since 8 digits are sufficient for single-precision. Integer values from single-precision waves are displayed using up to 16 digits.

Because the Integer column is formatted to display as integer, 3.141592653589793 is displayed as 3. If you clicked on Integer[1], the entry line would show the full precision.

The story is the same for Decimal[1]. The cell area of the table follows the column's formatting but if you clicked Decimal[1], the full precision would appear in the entry line.

The Hex and Octal columns are set to hexadecimal and octal display respectively. The hex and octal formats do not support the display of fractional data so Hex[1] and Decimal[1] display error messages instead of values. This is further discussed under **Hexadecimal Numeric Formats** on page II-257 and **Octal Numeric Formats** on page II-257.

Date Values

Dates and times are represented in Igor date format — as a number of seconds since midnight, January 1, 1904. Dates before that are represented by negative values.

A date can not be accurately stored in the data values of a single precision wave. Make sure to use double precision to store dates and times.

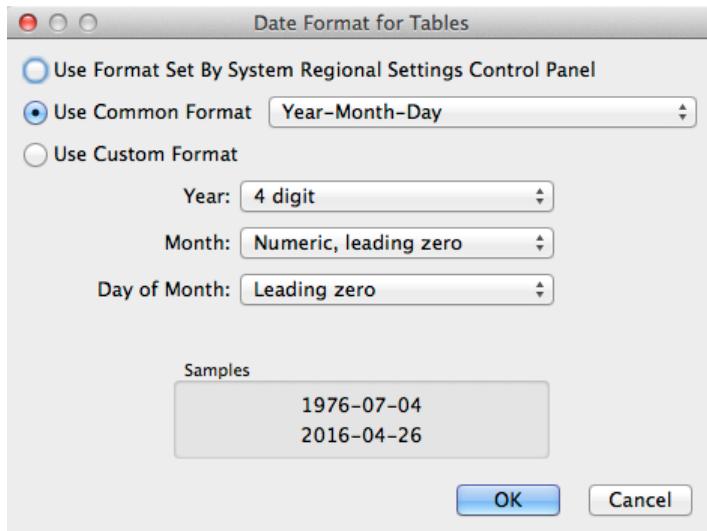
For further discussion of how Igor represents dates, see **Date/Time Waves** on page II-85.

The way you enter dates in tables and the way that Igor displays them is controlled by the Table Date Format dialog which you invoke through the Table menu. This dialog sets a global preference that determines the date format for all tables.

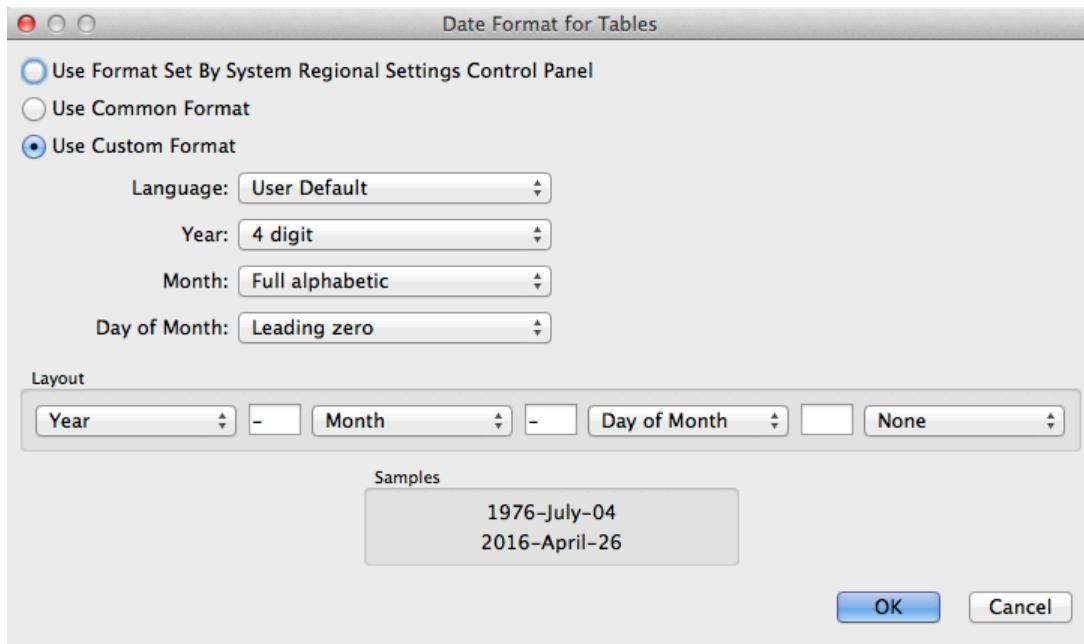
Chapter II-12 — Tables

If in the Table Date Format dialog you choose to use the system date format, which is the factory default setting, Igor displays dates in a table using the short date format as set by the Language & Region control panel (*Macintosh*) or by the Region control panel (*Windows*).

Alternatively, you can choose to use a common date format or a custom date format. Here is what the dialog looks like if the Use Common Format radio button is selected:



You can access even more flexibility in those rare cases where it's needed by clicking the Use Custom Format radio button:



When using a common or custom date format that includes separators (e.g., 10/05/99 or 10.05.99), Igor is lenient about the number of digits in the year and whether or not leading zeros are used. Igor will accept two or four digit years and leading zeros or no leading zeros for the year, month, and day of month. However, when using a format with no separators (e.g., 991005 or 19991005), Igor requires that you enter the date exactly as the format specifies.

When you enter a value in the first unused column in a table, Igor must deduce what kind of value you are entering (number, date, time, date/time, or text). It then sets the column format appropriately and interprets what you have entered accordingly. An ambiguity occurs if you use date formats with no separators. For