The Load Delimited Text operation always considers carriage return and linefeed characters to mark the end of a line of text. It would be quite unusual to find a data file that uses these characters as values. In the extremely rare case that you need to load a carriage return or linefeed as a value, you can use an escape sequence. Replace the carriage return value with "\r" (without the quotes) and the linefeed value with "\n". Igor will convert these to carriage return and linefeed and store the appropriate character in the text wave.

In addition to "\r" and "\n", Igor will also convert "\t" into a tab value and do other escape sequence conversions (see **Escape Sequences in Strings** on page IV-14). These conversions create a possible problem which should be quite rare. You may want to load text that contains "\r", "\n" or "\t" sequences which you do not want to be treated as escape sequences. To prevent Igor from converting them into carriage return and tab, you will need to replace them with "\\r", "\\n" and "\\t".

Igor does not remove quotation marks when loading data from delimited text files into text waves. If necessary, you can do this by opening the file as a notebook and doing a mass replace before loading or by displaying the loaded waves in a table and using Edit→Replace.

Quoted Strings in Delimited Text Files

Comma-separated values (CSV) text files can be loaded in Igor as delimited text files with comma as the delimiter. Here is some text that might appear in a CSV text file:

```
1,London
2,Paris
3,Rome
```

Sometimes double-quotes are used in CSV files to enclose an item. For example:

```
1,"London"
2,"Paris"
3,"Rome"
```

In Igor6 and Igor7, double-quotes in a delimited text file received no special treatment. Thus, when loading the second example, Igor would create a numeric wave containing 1, 2, and 3, and a text wave containing "London", "Paris", and "Rome". The text wave would include the double-quote characters.

In Igor8 and later, by default, the Load Delimited Text routine treats plain ASCII double-quote characters as enclosing characters that are not loaded into the wave. So, in the second example, the text wave contains London, Paris, and Rome, with no double-quote characters.

This feature is especially useful when the quoted strings contain commas, as in this example:

```
1,"123 First Street, London, England"
2,"59 Rue Poncelet, Paris, France"
3,"Viale Europa 22, 00144 Rome, Italy"
```

Prior to Igor8, Igor would treat this text as containing four columns, because double-quotes received no special treatment and comma is a delimiter character by default. Igor8 loads this as two columns creating a numeric wave with three numbers and a text wave with three addresses.

Because of previously-established rules regarding column names in delimited text files, if you specify that the file includes column names using the LoadWave /W flag, LoadWave interprets quoted text as column names even if the text is all numeric. For example, if you use LoadWave/W and the file contains:

```
"1","2","3"
"4","5","6"
```

LoadWave treats the first line as column names. However, if you use LoadWave/W and the file contains:

```
1,2,3
4,5,6
```

LoadWave treats the first line as data, not column names. So, if your file contains quoted strings, you must omit the /W flag if the file does not contain column names.