Function	Comment
ContourNameToWaveRef	Returns a reference to a wave displayed as a contour plot. Used with the output of ContourNameList.
ImageNameToWaveRef	Returns a reference to a wave displayed as an image. Used with the output of ImageNameList.
TraceNameToWaveRef	Returns a reference to a wave displayed as a waveform or as the Y wave of an XY pair in a graph. Used with the output of TraceNameList.
TagWaveRef	Returns a reference to a wave to which a tag is attached in a graph. Used in creating a smart tag.
WaveRefsEqual	Returns the truth two wave references are the same.
WaveRefWaveToList	Returns a semicolon-separated string list containing the full or partial path to the wave referenced by each element of the input wave reference wave.
ListToWaveRefWave	Returns a free wave containing a wave reference for each of the waves in the semicolon-separated input string list.

This simple example illustrates the use of the CsrWaveRef wave reference function:

```
Function/WAVE CursorAWave()
   WAVE/Z w = CsrWaveRef(A)
   return w
End

Function DemoCursorAWave()
   WAVE/Z w = $CursorAWave()
   if (WaveExists(w)==0)
      Print "oops: no wave"
   else
      Printf "Cursor A is on the wave '%s'\r", NameOfWave(w)
   endif
End
```

Processing Lists of Waves

Igor users often want to use a string list of waves in places where Igor is looking for just the name of a single wave. For example, they would like to do this:

```
Display "wave0;wave1;wave2"
```

or, more generally:

```
Function DisplayListOfWaves(list)
   String list                 // e.g., "wave0;wave1;wave2"

   Display $list
End
```

Unfortunately, Igor can't handle this. However, there are techniques for achieving the same result.

Graphing a List of Waves

This example illustrates the basic technique for processing a list of waves.

```
Function DisplayWaveList(list)
   String list             // A semicolon-separated list.

   Variable index = 0
   do
      // Get the next wave name
      String name = StringFromList(index, list)
      if (strlen(name) == 0)
```