```
            for(j=0;j<3;j+=1)
                rowIndex=triIndices[i][j]
                for(k=0;k<3;k+=1)
                    sphereTrianglesPath[outRowCount][k]=tripletWave[rowIndex][k]
                    sphereTrianglesSurf[outcount2][k]=tripletWave[rowIndex][k]
                endfor
                outRowCount+=1
                outcount2+=1
            endfor

            // Close the triangle path by returning to the first vertex:
            rowIndex0=triIndices[i][0]
            sphereTrianglesPath[outRowCount][0]=tripletWave[rowIndex0][0]
            sphereTrianglesPath[outRowCount][1]=tripletWave[rowIndex0][1]
            sphereTrianglesPath[outRowCount][2]=tripletWave[rowIndex0][2]
            outRowCount+=2          // Increment row count and skip the NaN
        endfor
    End
```

See Also

SphericalInterpolate, **Triangulate3D**, **ImageInterpolate** with keyword Voronoi

Demo

Choose File→Example Experiments→Analysis→SphericalTriangulationDemo.

SplitString

SplitString /E=*regExprStr str* [, *substring1* [, *substring2*,… *substringN*]]

The SplitString operation uses the regular expression *regExprStr* to split *str* into subpatterns. See **Subpatterns** on page IV-186 for details. Each matched subpattern is returned sequentially in the corresponding substring parameter.

Parameters

str is the input string to be split into subpatterns.

The *substring1*…*substringN* output parameters must be the names of existing string variables if you need to use the matched subpatterns. The first matched subpattern is returned in *substring1*, the second in *substring2*, etc.

Flags

/E=*regExprStr* Specifies the Perl-compatible regular expression string containing subpattern definition(s).

Details

regExprStr is a regular expression with successive subpattern definitions, such as shown in the examples. (Subpatterns are regular expressions within parentheses.)

For unmatched subpatterns, the corresponding substring is set to "". If you specify more substring parameters than subpatterns, the extra parameters are also set to "".

The number of matched subpatterns is returned in V_flag.

The part of *str* that matches *regExprStr* (often all of *str*) is stored in S_value.

Examples

```
// Split the output of the date() function:
Print date()
  Mon, May 2, 2005

String expr="([[:alpha:]]+), ([[:alpha:]]+) ([[:digit:]]+), ([[:digit:]]+)"
String dayOfWeek, monthName, dayNumStr, yearStr
SplitString/E=(expr) date(), dayOfWeek, monthName, dayNumStr, yearStr
Print V_flag
  4
Print dayOfWeek
  Mon
Print monthName
  May
Print dayNumStr
  2
Print yearStr
```