What's New in IDL 8.2.3

EXELIS
Authorized IDL Distributor

새로운 기능

BUBBLEPLOT

BUBBLEPLOT을 이용해 2-D Cartesian 좌표 체계로 표현된 데이터 위에 3-D의 버블 차트를 만들 수 있습니다. 버블 차트는 데이터의 규모(magnitude)가 (x, y)지점에 버블의 크기로 표현되는 것 이외에는 scatter plot과 유사합니다. 이 함수를 이용하여 주제 및 지도 위에 표출하는 예제를 확인해보십시오.

FILE_GZIP/FILE_GUNZIP

FILE_GZIP과 FILE_GUNZIP 루틴은 GZIP 파일 포맷의 압축과 압축 해제를 가능하게 합니다. 또한 추가적으로 메모리 버퍼에 입력 파일을 압축 혹은 압축 해제할 수 있습니다

FILE_TAR/FILE_UNTAR

FILE_TAR와 FILE_UNTAR 루틴은 TAR 파일 포맷과 경로의 아카이브 및 아카이브 해제를 가능하게 해줍니다. TAR 파일을 만들 때, GZIP 압축을 이용할 수 있는 옵션도 있습니다. FILE_TAR를 이용하면 파일의 출력 대신 데이터 스트림에 아카이브를 할 수 있고, FILE_UNTAR를 이용할 경우에는 데이터 스트림에 아카이브 해제를 할 수도 있습니다.

FILE_ZIP/FILE_UNZIP

FILE_ZIP과 FILE_UNZIP 루틴은 ZIP 파일 포맷을 이용하여 아카이브와 아카이브 해제를 가능하게 해줍니다.

IDLffVideoRead

IDLffVideoRead class는 다양한 포맷의 비디오 파일을 읽을 때 사용할 수 있고, 비디오 프레임, 오디오 샘플, 데이터 패킷을 읽을 때도 사용할 수 있습니다.

QUERY_VIDEO

QUERY_VIDEO는 파일의 읽기 과정 없이 비디오 파일의 정보를 가져옵니다.

READ_VIDEO

READ_VIDEO함수는 IDLffVideoRead를 포함하며, 비디오, 오디오, 그리고/혹은 비디오 파일로부터의 데이터 스트림 방법을 제공합니다.

WRITE_VIDEO

WRITE_VIDEO 함수는 IDLffVideoWrite를 포함하며, 비디오 그리고/혹은 오디오 스트림을 비디오 파일로 쉽게 저장하는 방법을 제공합니다.

TERMINAL_SIZE

TERMINAL_SIZE 함수는 IDL 출력에서 콘솔로 사용되는 행과 열의 크기를 설정하고 검색할 수 있습니다. 이 함수는 IDL 워크벤치에서 콘솔, 혹은 IDL 커맨드 라인을 위한 "tty" 터미널 쉘에서 모두 작동합니다.

VOLUME

VOLUME 함수는 데이터의 3-D 배열을 시각적으로 표현해줍니다.

ZLIB_COMPRESS/ZLIB_UNCOMPRESS

ZLIB_COMPRESS과 ZLIB_UNCOMPRESS 루틴은 DEFLATE 압축 알고리즘을 이용하여 IDL 수치적 배열을 압축 및 압축 해제시킬 수 있습니다. 또한 압축 수준을 선택할 수 있을 뿐만 헤더의 유형(no header, ZLIB, or GZIP) 또한 지정할 수 있습니다.

Update 부분

AXIS: Minor Tick Linestyle

IDLgrAxis와 AXIS graphics 함수는 모두 새로운 SUBGRIDSTYLE 속성을 가집니다. 이 속성은 사용자로 하여금 축의 보조 눈금(minor tick)의 linestyle을 조절할 수 있게 해줍니다. IDLgrAxis에서는 디폴트로 -1로 설정이 되어 있는데 이는 보조 눈금 스타일은 주 눈금 스타일을 따르는 것입니다. AXIS 함수에서는 실선인 0이 디폴트입니다.

Graphics: Output to SVG (Scalable Vector Graphics)

IDL new graphics는 현재 확장가능 벡터 그래픽(SVG) 출력을 지원합니다. SVG 파일 포맷은 웹페이지에 쉽게 임베디드되는 확장형 그래픽 생성에 널리 사용됩니다. 이 " 벡터" 포맷으로 사용자는 다양한 authoring 툴을 사용하여 그래픽을 수정할 수 있습니다.예를 들어, contour plot 위에 c.Save, 'contour.svg' 를 입력하여 간단히 SVG로의 파일 저장을 할 수 있습니다. 저장되고 나면, 웹브라우저에서 결과를 열 수 있습니다. SVG 출력에 대한 자세한 사항은 그래픽 저장 방법을 참고해 주십시오.

Text: Overline Symbol

TEXT graphics 함수는 표현된 문자 주변에 선을 그려주는 새로운 심볼 ₩overline{ } 을 추가했습니다. 예를 들면, 사용자는 다음과 같이 plot과 문자열을 구분할 수 있습니다.
myPlot = PLOT(/TEST, TITLE="$1/7 = 1.₩overline{142857}$", FONT_NAME='Symbol')

Automatic Gridding for IMAGE, CONTOUR, and SURFACE Functions

이전에는 만약 사용자가 IMAGE, CONTOUR, SURFACE와 같은 함수를 사용할 때 불규칙적으로 조정된 데이터를 입력하면 IDL은 에러를 생성했었습니다. 하지만 지금은 IDL이 자동적으로 데이터를 grid 합니다. 그리딩 방법에 대한 자세한 내용은 IMAGE, CONTOUR, 또는 SURFACE를 참고해 주십시오. 사용자는 또한 ::GetData 방법을 통해 그리드 된 결과를 불러올 수 있습니다.
예를 들어

```
; 좁은 골짜기를 포함하는 산재된 고도 데이터
ds = READ_ASCII (FILEPATH('irreg_grid1.txt', $ SUBDIRECTORY = ['examples', 'data']))
x = ds.field1[0,*]
y = ds.field1[1,*]
z = ds.field1[2,*]

; 불규칙한 공간 데이터의 Contour plot
c = CONTOUR(z, x, y, /FILL, $
POSITION=[0.18,0.15,0.83,0.85], $
ASPECT_RATIO=1, $
GRID_UNITS="meters", $
XTITLE='X (m)', YTITLE='Y (m)',
$ TITLE='Inlet Elevation (m)')
c1 = COLORBAR(TARGET=c, /ORIENTATION, $
POSITION=[0.86,0.2,0.88,0.8], FONT_SIZE=12, TEXTPOS=1)

; 원시 데이터 지역의 Overplot
s = SYMBOL(x, y, 'o', /DATA, /SYM_FILLED)
```

LIST::ToArray with DIMENSION Keyword

LIST::ToArray 방법에 DIMENSION 키워드가 생겼습니다. 만약 DIMENSION 키워드를 사용하면, 결과는 표시된 요소가 공간을 따라 연결되는 배열이 됩니다. 만일 사용자가 많은 수의 배열을 연결시켜야 하는데 그 전체 크기를 모를 경우(예를 들어, 파일에서 데이터를 읽을 경우) LIST::ToArray 를 사용할 수 있습니다. 단순 배열보다 월등히 빠릅니다. 자세한 내용은 LIST::ToArray 방법을 참고하세요.

INTERPOLATE with DOUBLE Keyword

기본적으로, INTERPOLATE 함수는 언제나 X, Y, 그리고 Z 인수로 주어지는 그리드 위치를 단일정밀도의 부동소수점 값으로 변환(비록 원본 데이터가 배정밀도라고 해도) 합니다. 이전과의 호환성 유지를 위해 이는 지속적으로 유지됩니다. 그러나, 사용자는 DOUBLE 키워드를 사용함으로써 배정밀도의 그리드 위치를 얻을 수 있습니다. 두 경우 모두 실제 간섭기법은 언제나 배정밀도 연산을 통해 이루어집니다.

!CONST: Added Imaginary Number "i"

!CONST 시스템 변수에 !CONST.I 를 사용하여 dcomplex (0,1) 값을 포함하는 새로운 “I”필드가 생겼습니다. 사용자는 이를 이용하여 real 값을 complex 값으로 쉽게 변환할 수 있고, 복잡한 연산을 수행할 수 있습니다.