

# What's New in IDL 8.2.2

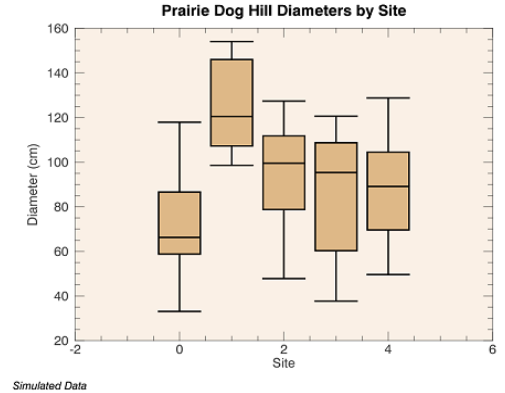
## 새로운 기능

### BOXPLOT

박스 형태를 만들고 whisker plot 기능이 추가되었습니다. 이를 위해서 [샘플의 최소값, 사분위수(四分位數, quartile) 중 낮은 값, 중간 값, 사분위수 중 높은 값, 샘플의 최대 값]를 필요로 합니다. 추가적으로 plot의 가외치(outlier)와 outlier value, 평균 값, 급수 등을 표현할 수 있으며, 앞서 언급한 5개의 값을 설정하기 위해서 새로운 함수인 CREATEBOXPLOTDATA를 이용할 수 있습니다.

### CREATEBOXPLOTDATA

boxplot를 지원하기 위한 함수입니다. 언급되었던 5개의 값 외에 데이터의 간격과 평균, 가외치, 가외치 추정값 등을 산출할 수 있습니다.

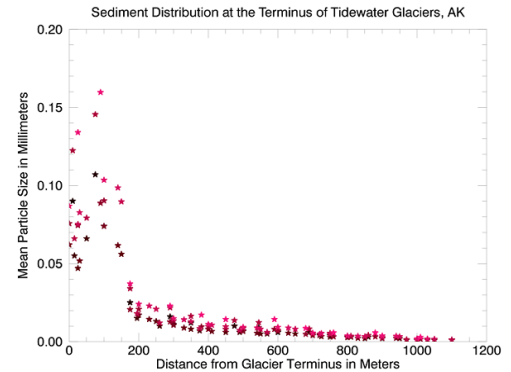


### IMAGE\_THRESHOLD

어떤 픽셀이든지 임계 값의 초과 여부를 판별하여 아래이면 0, 위라면 1이라는 2진 마스크를 반환하여 효과적으로 바이너리 이미지를 만들 수 있습니다. 임계값의 설정은 Isodata, Otsu, Moments, Maxentropy, Minerror, Mean법을 선택할 수 있습니다.

### SCATTERPLOT and SCATTERPLOT3D

새로운 scatterplot과 scatterplot3D 함수는 Cartesian 좌표를 이용하여 2차원 혹은 3차원의 공간에 점의 집합으로 데이터를 표시합니다.



### TIC and TOC

이 루틴은 함수나 프로시저로 호출할 수 있으며, 이를 통해 IDL의 실행시간을 프로파일할 수 있습니다. TIC이 프로시저로 쓰이면 TIC과 TOC 사이의 경과 시간을 알려줍니다. TIC이 함수로 쓰이면 TOC은 특정 호출에 대한 경과 시간을 알려주기 때문에 서브루틴이나 루프에서의 경과시간을 알 수 있습니다. 또한 IDL 코드 프로파일러의 활성화와 비활성화에 사용할 수 있습니다.

## Update 부분

### !CONST(시스템 변수)

새로운 시스템 변수 !const는 자주 사용되는 물리적 상수가 포함된 IDL 구조체입니다. 예를 들어, 수소 원자의 전이에 의해 방출되는 빛의 파장을 계산하는 Rydberg 상수도 포함합니다.

$$\frac{1}{\lambda} = R_{\infty} \left( \frac{1}{n_1^2} - \frac{1}{n_2^2} \right) = \frac{m_e e^4}{8 \epsilon_0^2 h^3 c} \left( \frac{1}{n_1^2} - \frac{1}{n_2^2} \right)$$

먼저, Rydberg 상수를 확인합니다.

```
PRINT, !CONST.me* !CONST.ev^ 4/ ( 8 * !CONST.eps0^ 2 * !CONST.h^ 3 * !CONST.c )
PRINT, !CONST.Rydberg
```

IDL prints:  
10973732.  
10973732.

n = 2를 적용하여 계산을 합니다.

```
PRINT, 1e9/(!CONST.Rydberg*(1 - 1./4)), " nm"
```

IDL prints:  
121.50227 nm

## RANDOM NUMBER GENERATOR

난수 생성 함수인 randomu와 randomn는 이제 Mersenne Twister algorithm을 사용합니다. 이는 이전의 Numerical Recipes의 난수 생성 알고리즘보다 개선된 알고리즘입니다. 새로운 알고리즘은 매우 큰 다항식을 가지고 최대 32비트의 정밀도로 균일하게 분포됩니다. 만일 이전과 동일한 난수 생성을 원한다면 ran1 이라는 키워드를 사용하면 됩니다.

## Graphics: CURRENT keyword

그래픽 함수에서 기존의 그래픽에 대한 참조를 전달할 수 있도록 current 키워드가 업그레이드 되었습니다. 이 키워드를 포함하는 그래픽은 현재의 그래픽 윈도우가 되고, 새롭게 생성된 그래픽은 그 윈도우에 삽입됩니다.

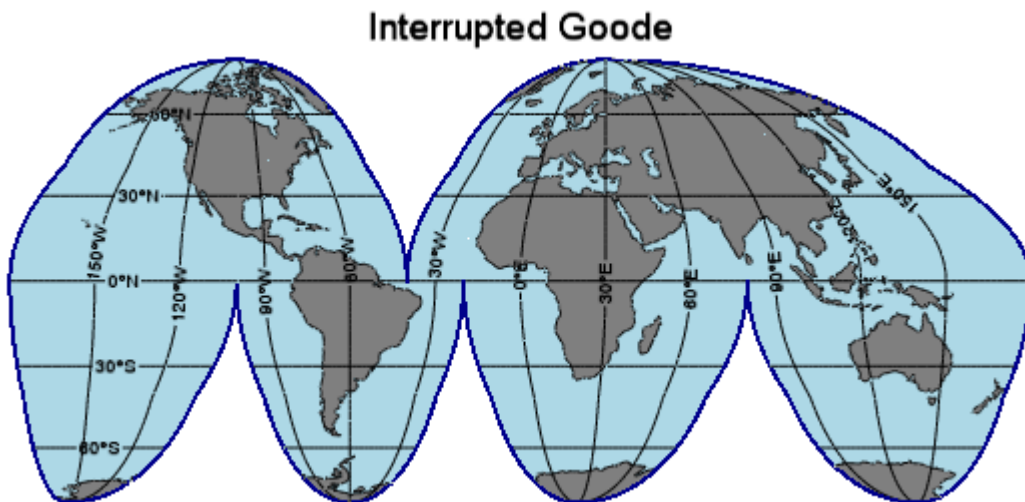
## Graphics: IMAGE with MAP PROJECTION keyword

표현하고 있는 이미지가 지도 투영이 된 것이라면, IDL은 자동적으로 이미지의 범위에 맵의 상한을 설정합니다. 이를 통해 수동으로 이미지의 범위를 계산하지 않고, 자동으로 box의 축을 표시할 수 있습니다.

## Graphics: MAP HORIZON

모든 지도 투영법에 맵 수평선을 표시할 수 있습니다. map과 map\_grid 함수는 세 가지 속성(horizon\_color, horizon\_linestyle, and horizon\_thick)을 가집니다. 예를 들면 다음과 같이 사용할 수 있습니다.

```
; Create a map using the interrupted Goode projection.  
mGoode = MAP('Interrupted Goode', FILL_COLOR='light blue', $  
  HORIZON_COLOR='dark blue', HORIZON_THICK=2, TITLE='Interrupted Goode')  
mc = MAPCONTINENTS(/FILL_BACKGROUND, FILL_COLOR='gray')
```



## Graphics: POSITION Property

이전 버전에서는 PLOT이나, CONTOUR 같은 '메인' 그래픽 함수에서 정확한 위치를 확인하기 위해 position 키워드를 설정하여 했었지만, 이제는 모든 그래픽 객체에 새로운 position 속성을 추가할 수 있습니다. Position은 [x1, y1, x2, y2]와 같이 네 개 요소의 벡터로 설정하며, 그래픽의 왼쪽 하단과 오른쪽 상단을 의미합니다. 생성 후, position은 2 혹은 4 개 요소의 벡터 중 하나를 설정할 수 있습니다. 이 중 2 개 요소가 설정된 경우는 그래픽 중심을, 4 개 요소로 설정된 경우는 position에 맞게 계산 및 조정됩니다.

## Graphics: Speed Improvement

모든 new graphics의 시작 속도를 개선했습니다. 추가적으로 plot함수의 속도는 현저히 빨라졌으며, 특히 많은 수의 포인트와 활성화 하드웨어 렌더링이 개선되었습니다. 예를 들어, 다음을 시도해 보십시오.

```

n = 1e6
x = RANDOMU(seed, n)
PRINT, 'Graphics functions:'
FOR i=0,2 DO BEGIN & TIC & p = PLOT(x) & TOC & p.CLOSE & ENDFOR
PRINT, 'Direct graphics:'
FOR i=0,2 DO BEGIN & TIC & PLOT, x & TOC & WDELETE & ENDFOR

```

IDL prints:  
Graphics functions:  
% Time elapsed: 2.1350000 seconds.  
% Time elapsed: 0.65100002 seconds.  
% Time elapsed: 0.64999986 seconds.  
Direct graphics:  
% Time elapsed: 0.84099984 seconds.  
% Time elapsed: 0.81699991 seconds.  
% Time elapsed: 0.81400013 seconds.

결과를 통해 많은 포인트를 plot하는데 있어 plot 함수(new graphics)가 다이렉트 그래픽스보다 속도가 더 빠름을 확인해 주십시오.

### IDLgrPlot: New HISTOGRAM styles

IDLgrPlot에서 histogram = 2라는 설정을 이용하여, 첫 번째 구간과 마지막 구간이 다른 구간과 같은 폭이 되도록 할 수 있습니다 (참고로, histogram = 1 설정에서는 x축 범위로 인하여 첫 구간과 마지막 구간이 다른 구간 폭의 절반만 표출됩니다). histogram = 3 설정은 히스토그램의 가로선의 시작을 x좌표로부터 다음 x좌표까지로 설정하게 됩니다.

### PLOT and BARPLOT: New HISTOGRAM Styles

PLOT과 BARPLOT 모두 수평선 혹은 수직선을 그리기 위한 새로운 히스토그램 속성이 생겼습니다. 수평 bar의 경우 각 x위치에서 시작하여 각 bin에 걸쳐있고, PLOT은 수평 bar가 각 x 위치를 중심으로 하는 수평선/수직선을 그릴 수 있는 새로운 '계단' 속성을 사용할 수 있습니다. 예를 들어 다음 코드는 동일한 히스토그램 데이터를 이용하여 각기 다른 4 가지 그래프를 표현합니다.

```

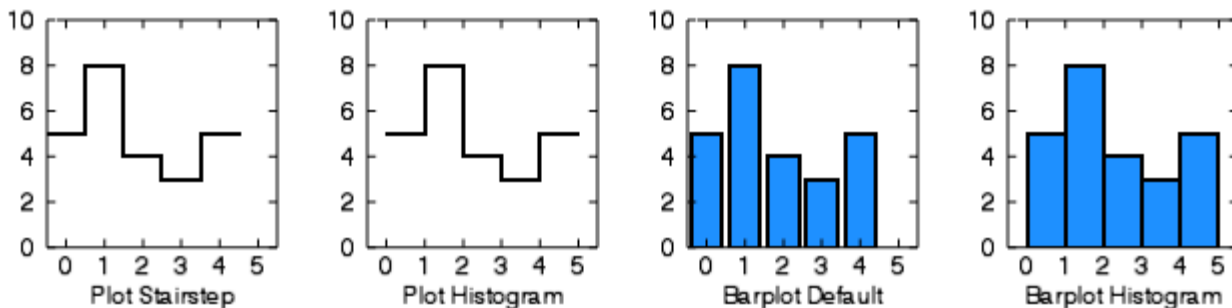
h = HISTOGRAM(5*RANDOMU(1,25), BINSIZE=1, MIN=0.0, LOCATIONS=loc)
ex = {THICK:2, XRANGE:[-0.5,5.5], YRANGE:[0,10], XMINOR:0, XTICKLEN:0.02, YMINOR:0, YTICKLEN:0.02}

```

```

p1 = PLOT(loc, h, /STAIRSTEP, LAYOUT=[4,3,1], _EXTRA=ex, XTITLE='Plot Stairstep')
p2 = PLOT(loc, h, /HISTOGRAM, LAYOUT=[4,3,2], _EXTRA=ex, /CURR,XTITLE='Plot Histogram')
p3 = BARPLOT(loc, h, LAYOUT=[4,3,3], _EXTRA=ex, /CURR,XTITLE='Barplot Default')
p4 = BARPLOT(loc, h, LAYOUT=[4,3,4], _EXTRA=ex, /CURR, /HISTOGRAM, XTITLE='Barplot Histogram')

```



### ROUTINE\_FILEPATH Function

만일 ROUTINE\_FILEPATH에 인수가 설정되지 않는 경우, 이 함수는 자동적으로 현재 범위의 루틴의 전체 경로를 반환합니다. 동일한 디렉토리에서 반복적으로 파일을 선택해야 하는 경우와 코드 안에서 루틴 이름을 하드코드하고 싶지 않을 때 유용합니다.

### Workbench Editor: Automatic Close of Begin/End Blocks

IDL 워크벤치 에디터에서, 'begin'이 들어간 문장의 끝에서 '엔터키'를 치면 자동적으로 'end'가 들어간 명령어가 입력됩니다. 이러한 기능은 모든 IDL 블록: IF/ENDIF, FOREACH/ENDFOREACH, FOR/ENDFOR, ELSE/ENDELSE, WHILE/ENDWHILE, CASE/ENDCASE, SWITCH/ENDSWITCH, 그리고 REPEAT/ENDREP을 지원합니다. 또한 입력한 대/소문자(대문자라면, 소문자라면, 첫문자만 대문자로)에 따라 'end'의 대/소문자도 동일하게 생성됩니다. 이 기능의 비활성화를 원하실 경우 환경설정 → IDL → 편집기에서 조절할 수 있습니다.

### Workbench Editor: Automatic Formatting of Pasted Code

IDL 워크벤치 에디터에서, 한 줄 이상이 들어간 블록을 '붙여넣기'를 할 때, 에디터는 자동적으로 들여쓰기와 붙여 넣은 서식을 적용합니다. 이는 수동으로 소스 → 포맷 항목을 선택하는 것과 같습니다. 이 기능의 비활성화를 원하실 경우 환경설정 → IDL → 편집기에서 조절할 수 있고, 또는 붙여넣기한 문장의 서식을 IDL → 포맷터에서 제어할 수 있습니다.