

GRAPHICS ENHANCEMENTS

Antialiasing:

By default, antialiasing is now used when drawing all lines, polygons, shapes, contours, and surface lines in Graphics. The `ANTIALIAS` property can be used to turn off the default behavior.

A new `ANTIALIAS` property has also been added to `IDLgrPolygon` and `IDLgrSurface`. The default value for `IDLgrPolygon` and `IDLgrSurface` objects is 0.

Colorbar and Legend:

You can now insert a colorbar and a legend from the toolbar in a Graphics window.

Color Tables:

- Custom color tables can now be created using the `COLORTABLE` function.
- Additional color tables based on ColorBrewer schemes are available.

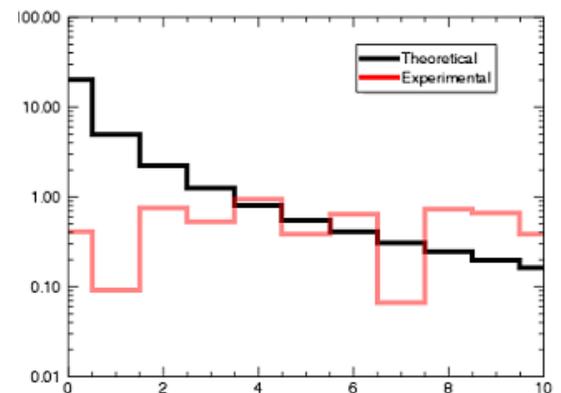
Graphics Event Handler:

You can now easily override the default event-handling behavior on graphics windows. The new `EVENT_HANDLER` property lets you pass in an object reference to an IDL class that can implement the following methods: `::MouseDown`, `::MouseUp`, `::MouseMove`, `::MouseWheel`, `::KeyHandler`, and `::SelectChange`. When one of these events is received by the graphics window, the appropriate method is called on your object first. Your object can then perform any necessary actions, and then return either 1 to continue processing the event, or 0 to swallow the event. The `EVENT_HANDLER` property is available on both the `WINDOW` object and the `WIDGET_WINDOW` function. See [Creating an Event Handler Class to Control Events](#) for details on how to write this object class.

PLOT HISTOGRAM property:

With the `HISTOGRAM` property, the `PLOT` function can now display data in a histogram style, with horizontal and vertical lines connecting each point. For example:

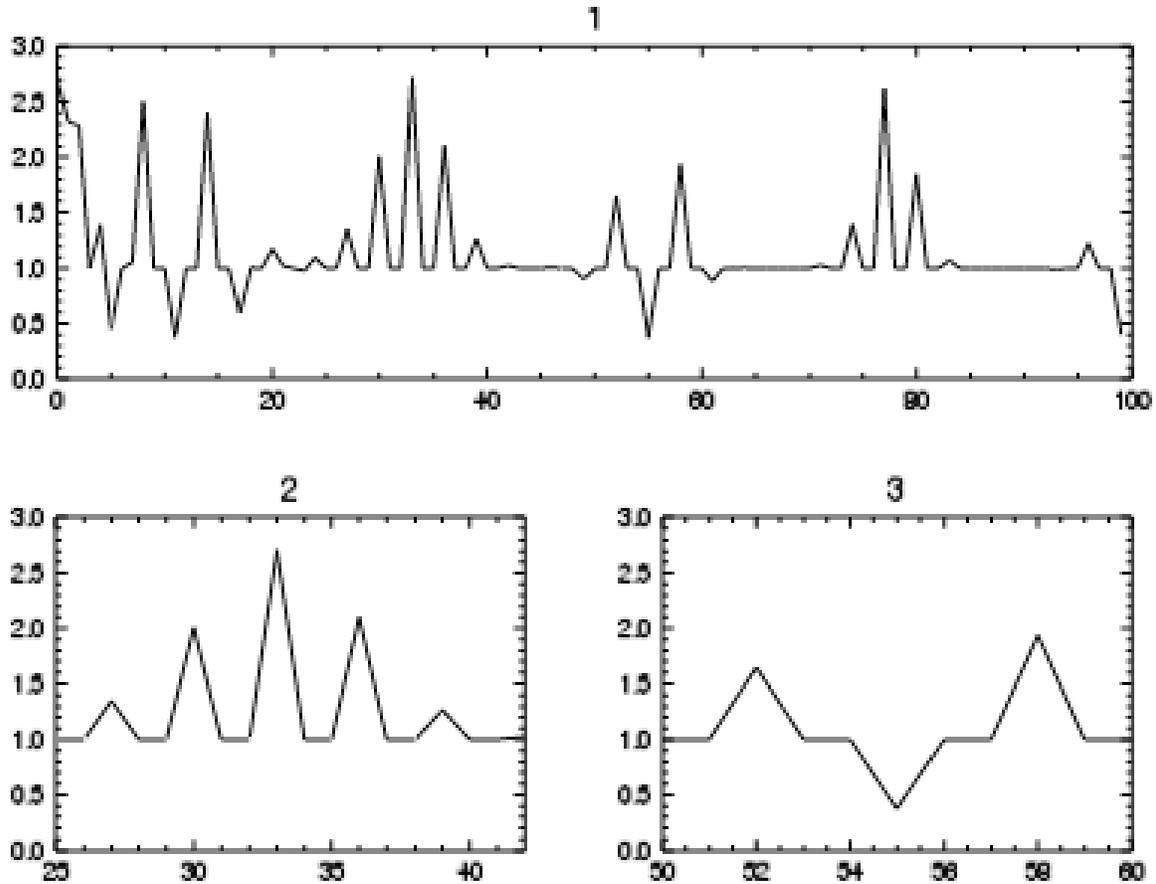
```
a = findgen(11)
b = 20./(a+1)^2
p1 = PLOT(a, b, '4', /HISTOGRAM, NAME='Theoretical', $
/YLOG, YRANGE=[0.01,100])
p2 = PLOT(RANDOMU(-1, 11), '4r', NAME='Experimental', $
TRANSPARENCY=50, /HISTOGRAM, /OVERPLOT)
h = LEGEND(POSITION=[9,50], /DATA)
```



See `PLOT` for more information.

Multiple Plot Layout :

When displaying multiple plots in a window, IDL now automatically calculates margins. You can specify just the order of the plots using the LAYOUT keyword, rather than specifying exact locations using POSITION, as shown below:



```
fn = exp(SIN(FINDGEN(100))^FINDGEN(100))
```

```
w1 = WINDOW()
```

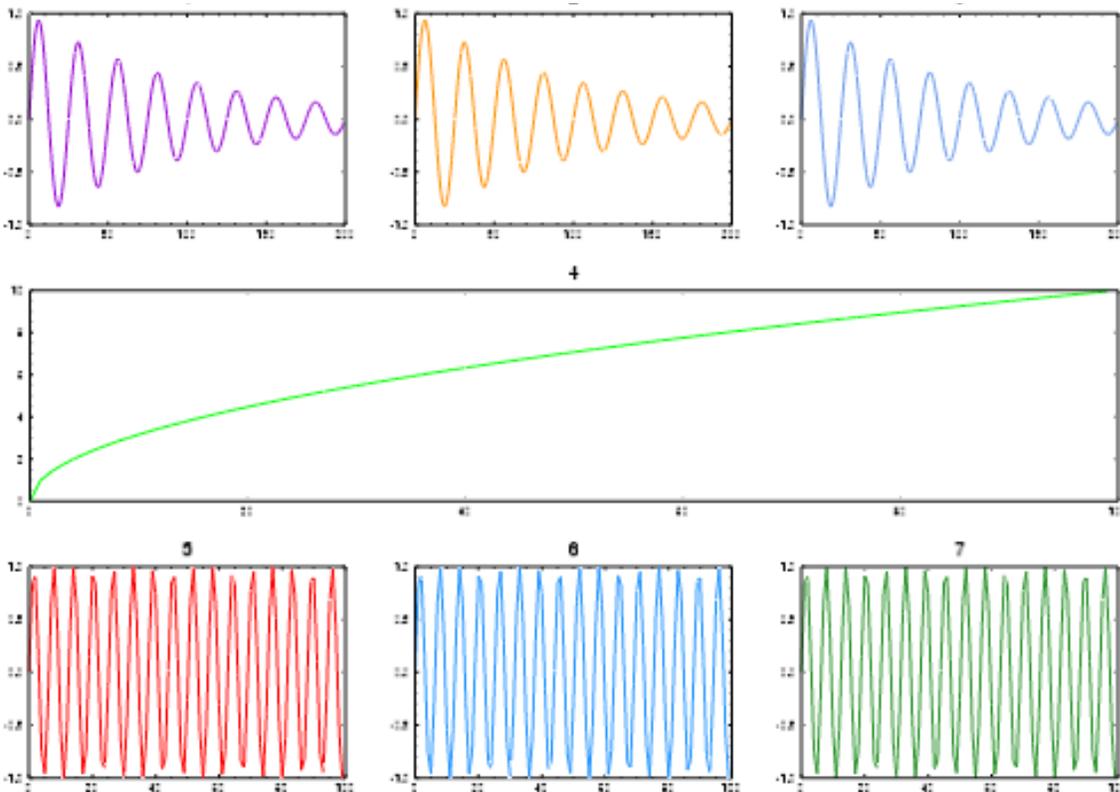
```
p1 = PLOT(fn, LAYOUT = [1, 2, 1], /CURRENT, TITLE = '1')
```

```
p2 = PLOT(fn, LAYOUT = [2, 2, 3], xrange=[25, 42], /CURRENT, TITLE = '2')
```

```
p3 = PLOT(fn, LAYOUT = [2, 2, 4], xrange=[50, 60], /CURRENT, TITLE = '3')
```

Multiple Plot Layout (continued):

The next example displays several plots in a 3x3 matrix in a single window:



```
fn = exp(SIN(FINDGEN(100)) ^ FINDGEN(100))
```

```
w1 = WINDOW()
```

```
p1 = PLOT(fn, LAYOUT = [1, 2, 1], /CURRENT, TITLE = '1')
```

```
p2 = PLOT(fn, LAYOUT = [2, 2, 3], xrange=[25, 42], /CURRENT, TITLE = '2')
```

```
p3 = PLOT(fn, LAYOUT = [2, 2, 4], xrange=[50, 60], /CURRENT, TITLE = '3')
```

Miscellaneous Graphics Changes:

- **Tick Format** - The AXIS_STYLE “exponent” value is now available to show exponential notation in plots.
- **Abbreviated Keywords** - IDL Graphics keywords can be abbreviated.
- **POLYGON** - Added the RGB_TABLE and VERT_COLORS properties to POLYGON
- **NCDF_CREATE** - Added of the NETCDF3_64BIT keyword to NCDF_CREATE

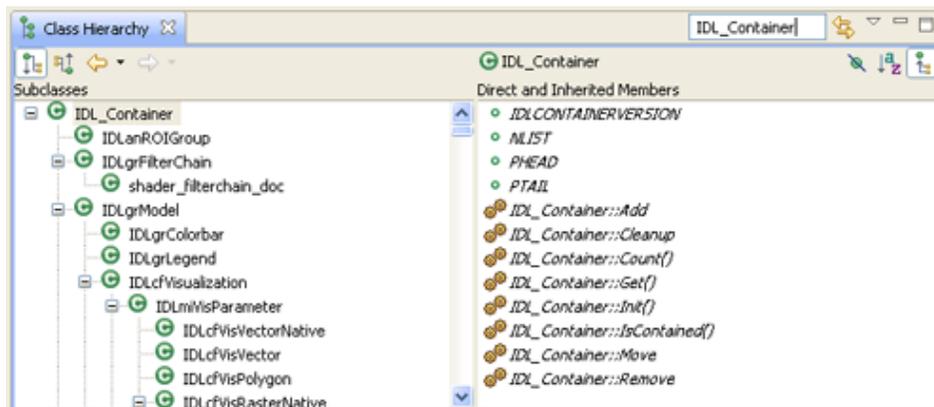
WIDGETS

- **Combobox** - New LIST_EVENTS keyword allows events to be generated when using the arrow keys to navigate combobox items.
- **Table Cell Events** - On Windows platforms, the ALL_EVENTS keyword now works as it does in UNIX platforms, returning cell events with keyboard arrow navigation.
- **Tree Scrolling** - Tree widgets can be scrolled using a mouse scroll wheel on UNIX as well as on Windows platforms.

WORKBENCH

Class Hierarchy Browser:

The Class Hierarchy browser is a view that displays the structure of the object hierarchy. To expose this feature, select **Window > Class Hierarchy**.



MISCELLANEOUS

- The IDLffJPEG2000 object has a new GML_ROOT_INSTANCE property that provides georeferencing information for JPEG2000 files.

Proleptic Gregorian Calendar:

The JUL2GREG and GREG2JUL routines let you convert from Julian dates to the proleptic Gregorian calendar and vice versa. The proleptic Gregorian calendar is produced by extending the Gregorian calendar backwards to dates preceding its introduction in 1582. In the proleptic Gregorian calendar, there are no missing days in October 1582, every 4 years is a leap year, except if the year ends in a "00" then it is not a leap year, unless it is also divisible by 400 (in which case it is a leap year).

The JUL2GREG and GREG2JUL routines let you convert from Julian dates to the proleptic Gregorian calendar and vice versa. The proleptic Gregorian calendar is produced by extending the Gregorian calendar backwards to dates preceding its introduction in 1582. In the proleptic Gregorian calendar, there are no missing days in October 1582, every 4 years is a leap year, except if the year ends in a "00" then it is not a leap year, unless it is also divisible by 400 (in which case it is a leap year).