

New features in APLX Version 3

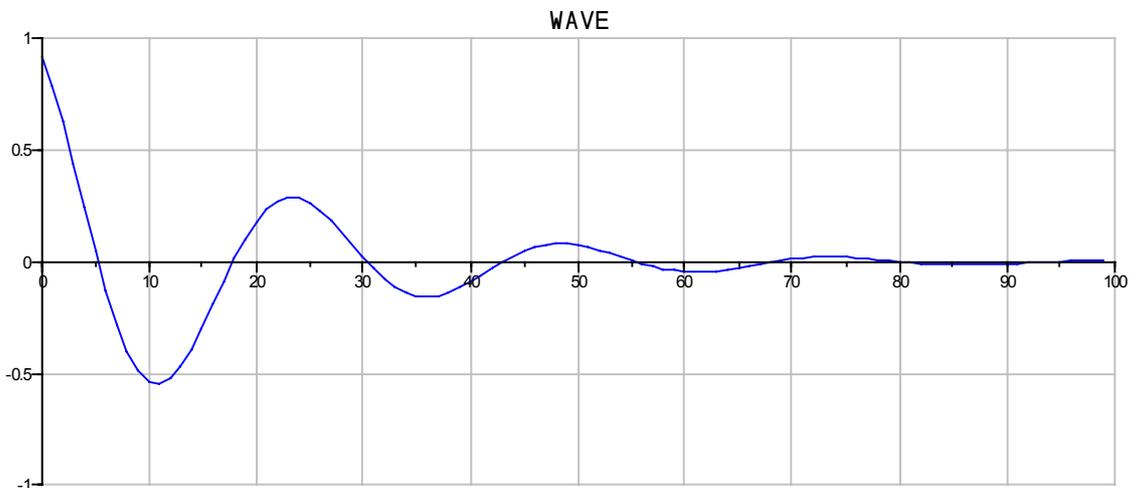
Built-in charting facility

APLX Version 3 includes a built-in, very powerful, but easy-to-use facility for producing business, statistical, and scientific graphs. You can use it in three ways:

- At its simplest, you can use a new 'right-click' menu item to bring up an instant chart of an APL variable. For example, if you type:

```
WAVE←(2÷0.25×⍋100)×*-0.05×⍋100
```

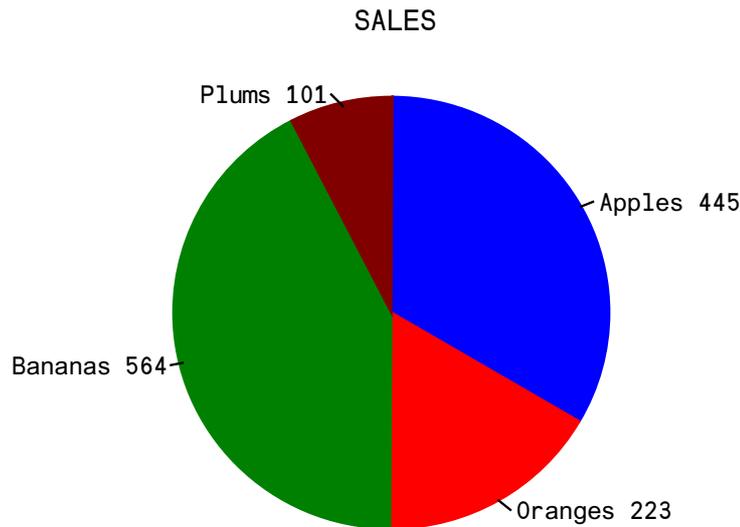
and then right-click on the variable name `WAVE`, there is now a new 'Display as Chart' item in the pop-up menu. (On the Macintosh, just click-and-hold instead of right-clicking). Selecting that item brings up a window containing an instant graph of the data:



If the variable is a matrix, APLX will make an intelligent guess about which rows or columns are the Y values and which (if any) the X values, but you can use the Chart menu to change these parameters, as well as selecting logarithmic scales or changing the type of the graph (for example, to a Scatter or Bar chart). If the chart is a nested matrix of strings and numeric values, the strings are used as labels of a bar chart. You can change this to a pie chart using the menu. For example, if you type:

```
SALES←2 4p'Apples' 'Oranges' 'Bananas' 'Plums' 445 223 564 101
```

and then use the pop-up menu item Display As Chart, the default will be to draw a Bar chart. You can use the Chart menu to change this to a pie chart:



- The new `⊠CHART` system functions allows the APL programmer to produce quick charts with almost no programming. The right argument is the data, and optional left argument allows you to specify how the data should be displayed. If you do not specify, APLX makes an intelligent choice for you. For example:

Draw a sine wave. Because no X values are specified, implicit X values of 0, 1, ..., 99 are used:

```
⊠CHART 1∘0.1×⍳100
```

Draw a sine wave with the X values specified in degrees. We provide a two-column matrix; APLX assumes the second column is the X axis, because it increments regularly:

```
⊠IO←1
⊠CHART (1∘(10×0,⍳36)×∘2÷360),[1.5]0,10×⍳36
```

Draw a bar chart:

```
⊠CHART X←2 4ρ'Apples' 'Pears' 'Oranges' 'Plums' 445 323 345 765
```

Draw the same data as a pie chart:

```
'type=pie' ⊠CHART X
```

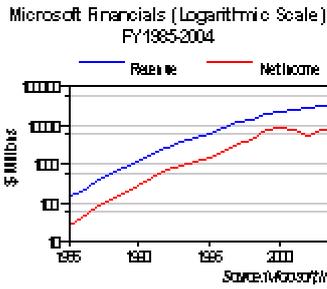
Draw some random data as two, named series, against implicit X values 0, 1, 2...:

```
'x=implicit' 'seriesname=US' 'seriesname=EU' ⊠CHART ?2 8ρ100
```

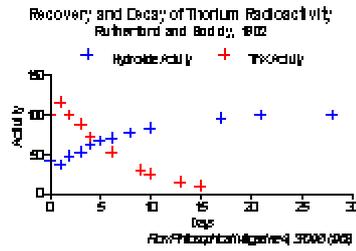
- For the most control over the appearance of the graph, you can use a new `⊠WI` object class, the **Chart**. This is a control which can display a range of different business, statistical and scientific graphs. The actual data which is displayed in the graph is held in one or more **Series** objects.

The Chart object can display the following types of graph:

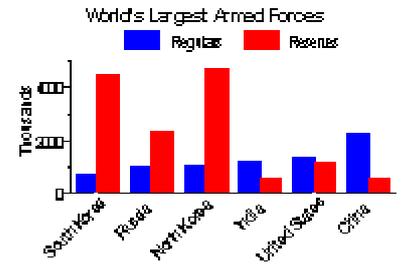
Line



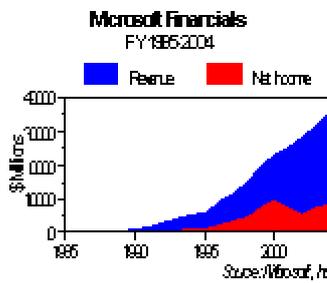
Scatter



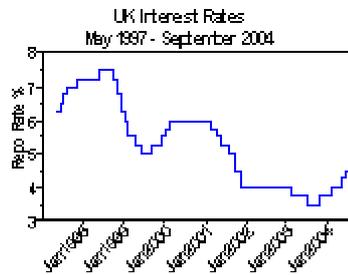
Bar



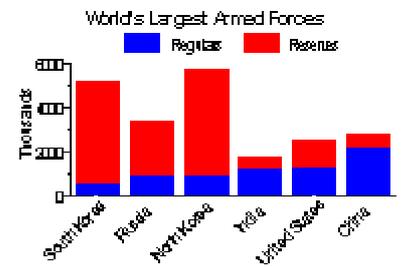
Area



Stair



Stacked Bar



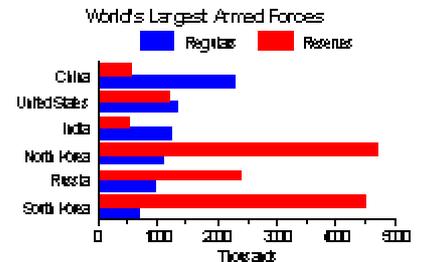
Hi Lo Open Close



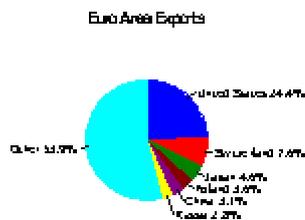
Candlestick



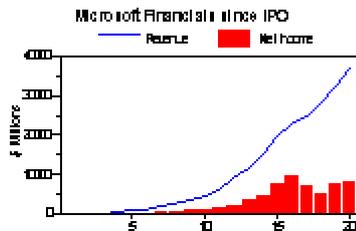
Horizontal Bar



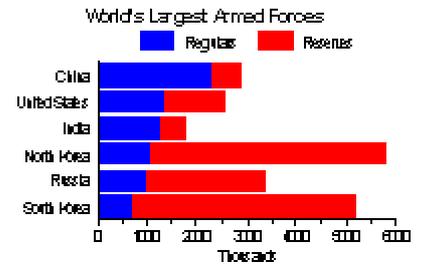
Pie



Mixed



Stacked Horizontal Bar



Although you can customize many aspects of the display of a Chart object, it is designed so that it automatically selects defaults which should give a good appearance for the chart with minimal programming. For example, if a chart is re-sized, by default the font sizes are adjusted appropriately, and reasonable values for the chart scale, tick positions, and labels are automatically chosen.

Charts by default are displayed in color. You can select your own colors for all the major elements of a chart, or you can leave the Chart object to select defaults. You can also ask for the chart to be displayed in monochrome.

As well as being displayed within your `GUI` windows, charts can also be printed, copied to the clipboard, and saved to file in various formats.

You can also customize a chart by adding your own graphics elements using the `Draw` method.

Image Manipulation

The new `GUI Image` object is an interface to ImageMagick™, which allows you to do image conversion to/from over 90 different graphics formats. These include:

<i>Tag</i>	<i>Description</i>
BMP	Microsoft Windows bitmap
EMF	Microsoft Enhanced Metafile (32-bit)
EPS	Adobe Encapsulated PostScript
GIF	CompuServe Graphics Interchange Format
JPEG	Joint Photographic Experts Group JFIF format
PDF	Portable Document Format
PICT	MacOS PICT file
PNG	Portable Network Graphics
SVG	Scalable Vector Graphics
TIFF	Tagged Image File Format

It also provides a huge range of image manipulation operations. For example, you can shear, scale, crop, rotate or skew the image, sharpen edges, adjust colors and intensity, and filter out noise. You can also apply special effects to the image, such as making it look like a charcoal drawing or an oil painting.

You can use the `Draw` method (or a Chart object) to create the image to be manipulated or saved, or you can read it from file or from the clipboard.

New Web Browser class

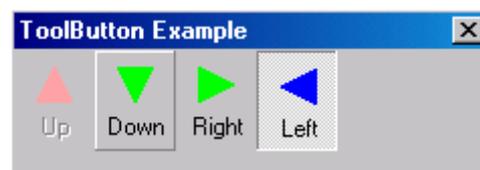
The new `Browser` object allows you to incorporate web pages in your APLX application. It is based on Internet Explorer under Windows, and Safari under MacOS X. (Under Linux, a limited version of the Browser object is provided).

New Networking classes

APLX Version 3 introduces four new `Browser` networking and Internet access classes. The `SendMail` and `GetMail` objects allow you to send and receive e-mail under program control, and the `HTTPClient` object fetches pages and other files from web sites into APL variables. There is also a new `Socket` object for low-level networking.

ToolButton Control

The `ToolButton` class is similar to a Button, but allows you to specify one or more images (as bitmaps) which can be displayed on the button face, depending on its state (Normal, Disabled, Being Clicked, and Selected):



It is typically used for implementing toolbars and similar controls.

It can optionally also display a text caption. In addition, it has built-in capability to act as a toggle button (like a Check box), or as one of a group (like a Radio button).

Other `Browser` enhancements

Other `Browser` enhancements include:

- The `tooltip` property implements hover-help on controls.
- The `ScreenToClient` and `ClientToScreen` methods convert from control coordinates to screen coordinates, and vice versa.
- The `PointToCell` method converts from coordinates of a Grid control on the screen to a cell position. It is particularly useful for responding to mouse-down events on the Grid, if you need to know which cell the user has clicked in.
- The Grid object can now display multi-line text in cells.
- The Form (or Window) object now supports the `bitmap` property, allowing you to retrieve the window image.

Built-in database interface: `DBSQL`

The new system function `DBSQL` provides a built-in, extensible, cross-platform interface to standard relational databases such as Oracle, SQL Server, IBM DB2, MySQL, or PostgreSQL, using Structured Query Language (SQL). It can also be used (via ODBC) to access data from spreadsheets, Microsoft Access, db2 files, and many other data sources. You can use it to retrieve data from relational databases as nested arrays in the workspace, or to update a database directly from within APL.

In this sequence, we are using the sample database supplied with IBM's DB2:

Connect to the database. The return code 0 0 0 0 means no errors occurred:

```
DBSQL 'Connect' 'aplxdabc' 'DB2'  
0 0 0 0
```

Retrieve the number of rows in the 'emp_photo' table:

```
(RC ERRMSG DATA)←DBSQL 'Do' 'select count(*) from emp_photo'  
RC  
0 0 0 0  
DATA  
12
```

No errors occurred, and the result was 12. We could retrieve all 12 rows immediately using the 'Do' keyword, but instead we'll prepare a database statement which allows more control over the retrieval, as well as allowing us to see the types and names of the columns:

```
DBSQL 'PREPARE' 'CS1' 'SELECT * FROM EMP_PHOTO'  
0 0 0 0  
DBSQL 'EXECUTE' 'CS1'  
0 0 0 1 SQL_CURSH200C4  
DBSQL 'DESCRIBE' 'CS1'  
0 0 0 0 EMPNO PHOTO_FORMAT PICTURE  
EMPNO PHOTO_FORMAT PICTURE  
CHAR 6 VARCHAR 10 BLOB 102400  
0 0 1
```

We'll fetch the first five rows of the result:

```
(RC ERRMSG DATA)←DBSQL 'Fetch' 'CS1' 5  
ρDATA  
5 3  
2 ↑[2]DATA  
000130 bitmap  
000130 gif  
000130 xwd  
000140 bitmap  
000140 gif
```

The third column of the result contains binary data (pictures of the employees). At this point, we could display the pictures using a Picture object in `OWI`.

Close the statement, and disconnect from the database:

```
(RC ERRMSG DATA)←DBSQL 'Close' 'CS1'  
(RC ERRMSG DATA)←DBSQL 'Disconnect'
```

Other new System Functions

- `□TR` - translate text to/from APLX internal representation
- `□SYMB` - return total size of symbol table, and number of symbols used
- `□WSSIZE` - return total size of workspace
- `□LIB` - return list of files in a given directory

New argument types in `□NA`

Double- and Single-Precision floating point data types, and Unicode characters, are now supported in `□NA` arguments and results.

MacOS File Paths

Under MacOS, you can now use Unix-style path names (using / slash as a delimiter), as an alternative to the traditional Macintosh colon-delimited style. This applies to all places in APLX where file paths are used.

MacOS File Filters

The OpenFile and SaveFile dialogs under MacOS now accept filters based on file extensions as well as filters based on MacOS file types.

Platform compatibility notes:

APLX Version 3 runs under MacOS X, MacOS 9, Windows XP, 2000, ME, NT, 98, and 95, and all main Linux distributions issued since 2000 including RedHat 6.0 or later, Mandrake 8.1 or later, Debian 3.0 or later, and SuSe 8.1 and later. However, functionality may be restricted on some platforms, as follows:

MacOS: Although APLX Version 3 runs under both MacOS 9 and MacOS X, the following new features are available only under MacOS X: The Image, Browser, and Socket objects, and `□SQL`. Because MacOS 9 does not support text drawn at arbitrary angles, Charts may display less satisfactorily under MacOS 9. Users should note that support for MacOS 9 may be withdrawn in the next main release of APLX. The Image object requires ImageMagick™ 6.1 or later to be installed (available free of charge). `□SQL` requires iODBC to be installed (supplied with MacOS X).

Linux: The Browser object is implemented in a very restricted form under Linux. The Image object requires ImageMagick™ 6.1 or later to be installed (included on most Linux distributions, or available free of charge). `□SQL` requires unixODBC to be installed (included on most Linux distributions, or available free of charge).

Windows: The Image object requires Windows 2000 or XP. The GetMail, SendMail, HTTPClient and Socket classes may not run on some Windows 95 and Windows 98 systems. The Image object requires ImageMagick™ 6.1 or later to be installed (available free of charge).