



APL Product Watch

An overview of new and recently-updated APL products

The text of APL Product Watch is supplied by the product vendors.

News from IBM Corporation

New Features for Workstation APL2

Two Upgrades Shipped in 1999

TWO SERVICE UPGRADES to the APL2 workstation product set were shipped in 1999. In April, Level 4 for Windows and Level 15 for OS/2, AIX and Sun Solaris was produced. In November, Level 5 for Windows and Level 16 for OS/2, AIX and Solaris was made available. These two upgrades contained a number of enhancements, as well as fixes for user-reported problems. Among the new items:

- Files as Arrays
- Workspace compare tool
- Support for new data types in AP 127
- REXX Processor
- AP 488 (GPIB)
- Debugging features for the Object Editor
- New AP 145 commands
- New bitmap APL font
- Windows shortcut keys

For detailed information on these items, check the APL2 web page ("www.ibm.com/software/ad/apl"), see the README or "Service Information" files shipped with the upgrades, or contact us directly.

OS/2 and Windows users can download service from our Web site. AIX and Solaris users can open a problem report with the IBM Support Center to obtain the latest maintenance.

File Access Made Easy

The "Files as Arrays" facility, Processor 12, was first made available to mainframe APL2 users in APL2 Version 2 Release 1 (1991). With Service Level 5/16, it is now part of all of the currently-supported APL2 systems.

Processor 12 provides a new and unique way to access files from APL2. The `QNA` system function is used to associate an APL name with a file. Once that is done, you simply use standard APL language constructs with that name to access the items in the file. Catenation is used to add new items, and existing items

can be selected and modified with index, pick, and other primitives.

Two types of files are supported: standard operating system text files and APL files with arbitrary arrays stored as items in a vector. Text files may be accessed using fixed or variable record lengths.

Complete documentation on Processor 12 and how to use it is found in on-line documentation shipped with the upgrade. On Windows and OS/2, it's found in the APL2 User's Guide in the chapter "Associated Processors." On AIX and Solaris, a new documentation file, `README.p12`, has been added to the `/examples/qna` directory of the APL2 product.

Debugging with a Click

The built-in Object Editor supplied with APL2 on OS/2 and Windows provides users with GUI editing capabilities such as cut and paste, split and join, customization of colors and keys, and of course support for the APL character set.

In Service Level 4/15, debugging features were added to the Object editor. Two new menubar items were added: "Break-points," which allows you to set and clear stops and traces on the function you are editing, and "Signals," which provides the ability to immediately stop and resume execution. You can also set and clear stops and traces with a mouse click in a new prefix area on the left side of the edit window. In addition, a new session manager option opens the editor automatically, and places the cursor on the suspended line, when a stop or interrupt is encountered.

We hope to add additional debugging capabilities in the future. Let us know what you'd like to see.

APL2: Ready for Windows 2000

The APL2 for Windows product has been tested on beta versions of Windows 2000. For the most part the test was successful, but several minor problems were identified. Fixes for all but one of the problems have already been shipped, in Service Level 5. The remaining problem affects performance of text display (in AP 124 screens and the Object Editor). A fix for that problem will be shipped in the next service level. ■

Find APL2 on the World-Wide Web:
<http://www.ibm.com/software/ad/apl>

News from Soliton Associates

SOLITON ASSOCIATES ANNOUNCES AVAILABILITY of the beta release of the Java Interface (SJI) for SHARP APL for Linux systems. The SJI enables SHARP APL to call Java and for Java to call APL. The SJI enables an APL programmer to use all of the Java GUI classes as well as other Java classes available with Java, such as JDBC and Servlets.

The beta release of the SJI for Linux SHARP APL systems can be downloaded from www.soliton.com/linux and is available royalty free for personal use licenses.

SJI Architecture

The architecture of the SJI is simple, consisting of an APL toolkit and a SHARP APL Java Engine. The communication between the toolkit and the Java Engine is socket-based. The Java Engine is a Java program that can be executed in any Java environment, for example those available for Windows, NT, Unix, Linux and Mac/OS. The Java Engine is small and can also be executed as part of an applet within a browser (Netscape or MS-IE). This architecture is consistent with the server nature of SHARP APL. A typical user environment would consist of a SHARP APL server and a network-connected client on a Windows platform.

The Java engine can either call in to APL or be called by APL to initiate a connection. The Java engine can create objects of any Java class and invoke any method on such objects. The APL toolkit instructs the Java engine to create forms, buttons, toolbars, JDBC objects, etc. On receiving information from a user interface or another process, the Java engine sends information to APL.

The SJI enables an APL application programmer to refer to Java classes and methods using their Java names. Using this architecture allows the APL programmer access to new classes and methods without changes to the interface. The interface handles conversion of APL data structures into Java data structures and vice versa.

Features and Benefits

GUI Geometry

Because Java is device independent, it handles the geometry of GUI elements (buttons, text boxes, labels) by layering frames in a similar manner to HTML forms. The result is that resizing is automatically handled by Java, without having to count pixels.

Distributed Processing

Processing can be offloaded from the APL server by doing work on the client in Java, such as data validation.

Application as URL

Coupling the SBI (Soliton's HTTP server) with SJI, APL-Java applications can be accessed through their Internet address or URL. By specifying the URL associated to an SJI application, the SBI will send the Java engine to your browser as an applet and the application will communicate with the applet to create forms. This eliminates the need for DLLs or executables to be installed at the client.

Java calls APL

A Java application can call APL by way of some specific Java classes that are part of the SJI. For example, a J-Application Builder may be used to design a user interface and call APL for data processing and retrieval. The conversion of data structures between Java and APL is automatic.

Performance

The SJI buffers most Java commands to minimize communication between APL and the Java Engine. In most cases, only the "Submit" button will cause a TCP/IP transfer to the server, while the Java client will process buttons and fields input locally. To facilitate the work of GUI developers, the SJI can work in diagnostic mode where commands are not buffered but executed at once by the Java Engine.

Beta Limitation

With the current beta release of the SJI you can create GUI and JDBC applications where APL runs on a Linux server and Java executes on a Linux, Windows or MAC client. Only connecting out to the Java Engine is supported. The "Application as URL" requires the SHARP APL Browser Interface, which is not yet supported.

Contact Information

For up to date information on the SJI and SHARP APL for Linux, visit our website at "www.soliton.com/Linux". For commercial licenses, please contact Laurie Howard (ljh@soliton.com) at +31 20 646 4475 (Europe) or Nancy Lamb (nancy@soliton.com) at 716-256-6466 (North America).

For information about personal licenses for SHARP APL for Linux, please contact linux@soliton.com. ■

To submit your own APL product announcements to
APL Product Watch, contact Ray Polivka:
"Polivka@ACM.org".