CLOSE-UP: What had to happen to make Java and Cobol work together?

Given Netron's Connectivity Framework™ (available in Netron Fusion)— surprisingly little. Both Cobol and Java speak TCP/IP. Netron Frameworks use the Internet protocol to link Java clients on remote workstations to Cobol servers. A Cobol router program determines which Cobol/Unix servers are needed and signals them to store and retrieve data from RSL COM's 40 gigabytes of distributed Informix databases.

The only hiccup is the fact that Java knows nothing about fixed-length strings and Cobol knows nothing about any other kind. The development team solved that problem by framing Java utility classes that package strings into various fixed-length formats—character, time, date, integer, decimal, etc. Once these classes were framed, Java and Cobol meshed together smoothly.

Armed with generic Java-Cobol linkage frames, the designers embedded them in three-tier client/server frameworks that contain additional Java frames for building clients (either as applets or object groups) and Cobol frames for building servers. For example: the framework for Batch Submissions implements a Java interface to provide users with the ability to trigger certain jobs on the server machine and control their execution by passing the appropriate number of parameters to them. The server jobs can vary in function and may include Cobol programs for functions such as data mining and reporting.

Developers were guided by "wizards" that customize and invoke the Java-Cobol frameworks in order to design and build maintenance programs. These programs can maintain a parent entity and zero or more child entities within a single multi-tabbed interface. Information from the parent entity is always displayed while information about each child is presented through the appropriate tab. The frameworks automate the assembly of programs and objects to implement an unlimited variety of interaction scenarios.