Welcome to the Legacy Integration Supplement

By David McGoveran
Sr. Technical Editor &
Consulting Industry Analyst

What, exactly, does "legacy integration" mean? Most old hands in IT would certainly agree that integrating a non-mainframe application with a mainframe application would qualify. Of course, our interpretation of legacy integration has evolved over the years. We now include the integration of almost any applications, as long as the architecture of one of the applications is significantly older. Any application that was not designed to support the distributed architecture *du jour*, let alone the many standards *de facto* and *de jure*, is now said to be a legacy application.

We sometimes forget that early integration efforts were primarily batch file transfers or the later use of remote job entry emulation, but I can personally attest to some of this history. In the late '70s and early '80s, I used these and other techniques to integrate Unix applications running on DEC PDP-11 and then VAX-11 systems with various mainframe applications. Tedious to develop and slow, the approach was predestined to be replaced. As early as 1983, I began using a shared database machine to facilitate application integration among minicomputers and mainframes (with self-describing data formats, I might add) and then moved to asynchronous message processing in 1985. Although middleware standards and products now make such efforts easier and enable a broader range of integration solutions, the underlying technical issues and their solutions have not changed very much.

In a sense, the legacy adjective has nothing to do with deployment platform. Older applications typically represent an integration challenge, regardless of the type of deployment platform. They were written either for efficiency or to take advantage of proprietary (possibly obsolete) platform features. Tight coupling fostered efficiency, but in the extreme, leading to brittle, monolithic applications. Taking advantage of undocumented side-effects was rarely considered a crime. And speaking of documentation, even the most disciplined IT shops have lost or failed to produce some critical documentation. Almost by definition, legacy applications aren't victims of code bloat, let alone "self-documenting" code!

In this supplement, we focus on legacy integration insofar as it involves mainframe applications. There are many good reasons that mainframe applications represent a special challenge to integration efforts, or perhaps better, that the encroachments of integration efforts represent a special challenge to mainframe users. Mainframe performance, security, local resource, transaction management, and so on, have always been more carefully managed on the mainframe than other platforms. Furthermore, the mainframe offered more robust facilities for that purpose than were generally available on so-called "open" platforms. Those same properties made the mainframe environment less attractive for cheap, distributed computing architectures, so that distributed integration technologies have tended to develop outside the mainframe first.

Today, the gap between the native architectures of legacy applications and integration technologies has widened. At the same time, mainframe vendors have helped create integration standards that enable powerful options for integration with those technologies. Nonetheless, legacy integration remains for the foreseeable future a critical challenge for business integration. Web deployment, Web Services, XML, SOA and BPM have each brought new issues to light, while raising the expectations of business users regarding integration.

In our legacy integration supplement, we bring you the views of experts on most of these issues. **Michael Rawlins**, author of *Using XML With Legacy Business Applications*, provides a guide to legacy application support for XML. Well-known to many of our readers and the author of several books on integration, **David Linthicum** teaches the essential concepts for building an SOA involving legacy systems and using Web Services. **Mary Shacklett's** interviews of IBM expert Jim Rhyne, Forrester analyst Phil Murphy, and three customers about Web-based delivery of mainframe resources explore several alternatives to Web Services. In future issues, we're certain to cover the BPM aspects of legacy integration.

To all this, we've added two sets of interviews that will provide insight into this market and associated technologies. First, we are fortunate to have obtained the views of IBM's **Jim Rhyne**, distinguished engineer, eServer Tools Technology and Enterprise Modernization, and **Scott Cosby**, program director, WebSphere Business Integration. Second, several industry analysts have given us their views on legacy integration, including **Dale Vecchio** of Gartner, **Phil Murphy** of Forrester Research, **Michael Thompson** of The Butler Group, and **Nathaniel Palmer** of Delphi Group.

We've attempted to keep this supplement from getting too technical, while providing a balanced introduction to some of the key concerns and issues of legacy integration as they pertain to the mainframe. We hope this will help you find your way to success if you face the many challenges of integrating mainframe and non-mainframe applications. **bij**