

Realtime
publishers

Cloud-Enabling Legacy and Mainframe Applications
The Essentials Series

Connecting Legacy Application Data to the Cloud: Common Mistakes and Misconceptions

sponsored by



Don Jones

Introduction to Realtime Publishers

by **Don Jones, Series Editor**

For several years now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We’ve made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book’s production expenses for the benefit of our readers.

Although we’ve always offered our publications to you for free, don’t think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you \$40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the “realtime” aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We’re an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I’m proud that we’ve produced so many quality books over the past years.

I want to extend an invitation to visit us at <http://nexus.realtimepublishers.com>, especially if you’ve received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you’re sure to find something that’s of interest to you—and it won’t cost you a thing. We hope you’ll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Introduction to Realtime Publishers..... i

Connecting Legacy Application Data to the Cloud: Common Mistakes and Misconceptions.. 1

 Legacy Apps: Expensive, Mission-Critical, and Change-Averse..... 1

 The Top-Down Approach: One Solution to Rule Them All 2

 The More Moderate Approach: Modifying Your Mainframe Apps..... 3

 A Popular Approach: Adding Web Services Capabilities to Your Mainframe 4

 A Wish List for Cloud-Connected Legacy Data..... 6

Copyright Statement

© 2011 Realtime Publishers. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers its web site sponsors. In no event shall Realtime Publishers or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at info@realtimepublishers.com.

Connecting Legacy Application Data to the Cloud: Common Mistakes and Misconceptions

You've decided that you need to connect your mainframe data to the cloud. The previous article in this series outlined several reasons why you might want to do so, and now you're ready to find out how. In this article, we'll explore common, traditional approaches—and their downsides. We'll also look at strategic, tactical, and technical mistakes that companies make when heading down this path so that you can avoid them in your own organization.

Legacy Apps: Expensive, Mission-Critical, and Change-Averse

First, let's acknowledge that extending a mainframe application in any way can be a frightening proposition. These applications simply aren't *designed* for extensibility. In some cases, business logic is contained entirely within text-based user interfaces, making it difficult to automate data entry and retrieval without bypassing all of that business logic.

Mainframe applications typically sit right at the heart of the business. They're one of the most mission-critical aspects of the business' technology infrastructure, running the applications that make the business *work*. Messing with that kind of critical component is extremely risky: One wrong move, and you're literally out of business.

Let's look at three traditional approaches that all seek to extend the mainframe application data. For each, we'll examine the details and downsides, and see which ones meet the most important business criteria for this kind of project. Typically, businesses are looking for a solution that

- is quick and can be implemented for a specific project. Companies are rarely looking for some kind of unified, do-it-all solution, and instead tend to be driven by a specific, time-sensitive need.
- is inexpensive, at least compared with whatever alternatives are available.
- involves as little risk to the business as possible.
- is as reusable as possible so that when future projects come down the road, companies don't have to start this entire process over again from scratch.
- won't negatively impact mainframe performance because the mainframe occupies such a critical role within the business.

Let's see how three traditional approaches measure up.

The Top-Down Approach: One Solution to Rule Them All

With this approach, companies seek to figure out everything they might need to integrate their legacy data into, and to build a solution—either based on the mainframe itself, or integrated from a PC-based server—that will meet all of those needs. On paper, this solution is elegant: Solve the problem once, in a consistent and well-planned way. This might involve creating an entire adjunct to the mainframe application that is capable of communicating with external applications via some data-interchange protocol. Ensure that all the necessary business logic is built-in to that connectivity application, and enable it to access every aspect of the mainframe data.

In reality, this approach is generally one of those projects that never really get off the ground. To begin, it's a massive undertaking and will generally involve extensive discovery. You'll need to determine the full extent of the data that will require integration, and you'll need to consider all the potential ways in which someone might want to integrate with it from external systems. You'll likely be duplicating a great deal of business logic, giving you another whole mainframe application that has to be maintained (which is certainly a non-trivial concern in and of itself).

It'll be such a massive, all-hands effort that executive commitment will usually be required, making it much harder to get the necessary approvals to get the project underway. It'll be a high-profile project, with a lot more eyes on what's happening and a lot more red tape and process to slow things down. You'll be messing with the mainframe itself, which is rightly a cause for concern and caution.

The worst aspect of this approach is that it'll really hold up any specific projects that need this kind of integration. Although your programmers and architects design and build this “unified integration system,” projects that desperately need to access just a few bits of data will simply have to wait—jeopardizing business opportunities and holding back the organization as a whole.

In terms of the common-sense business criteria, how does this “top-down” approach measure up?

Criteria	Top-Down Approach		
Time to implement for a specific project	Lengthy		
Expense	High		
Business risk	High		
Reusability	Very good, if well-designed		
Mainframe performance impact	Moderate to high		

That’s not looking like an ideal approach.

The More Moderate Approach: Modifying Your Mainframe Apps

A more moderate approach is to simply focus on the project at hand. Rather than trying to create a one-size-fits-all integration layer to the mainframe app, you simply modify the mainframe app to accommodate the most immediate need.

This approach is certainly not without its downsides. In a best-case scenario, where you own the source code to the application and have experienced developers to do the work, you’re looking at a time-consuming project that entails an incredible amount of business risk—you are, after all, modifying an application that sits in the most mission-critical position possible. A less-than-best-case scenario will require you to go to the application vendor for modifications or to otherwise outsource the project—which will be expensive. Even if you’re using your own development resources, you’re likely looking at a long wait because most companies have a significant project list stacked up for their mainframe programmers.

If well-written, the modifications can often avoid negative performance impact because they'll be integrated directly with a native application. However, this approach also offers a lower return on investment (ROI) because this approach is project-specific. The next project that comes along will require its *own* modifications. Eventually, you'll have a heavily-modified application that's been tweaked to meet the needs of each project that comes along. That kind of heavily-customized application carries its own significant business risks in terms of long-term support and operational costs.

How does this approach meet our business criteria?

Criteria	Top-Down Approach	Modify the Mainframe App	
Time to implement for a specific project	Lengthy	Moderate to Lengthy	
Expense	High	Moderate to High	
Business risk	High	High	
Reusability	Very good, if well-designed	Poor	
Mainframe performance impact	Moderate to high	Low to Moderate	

Not so well. It's still time-consuming, incredibly risky, and even with your own developers doing the work on your own source code, it's an expensive proposition with poor reusability.

A Popular Approach: Adding Web Services Capabilities to Your Mainframe

Another approach, one made popular in recent years by vendors who play in this space, is to add a Web Services layer to the mainframe itself. Doing so turns the mainframe into a standards-compliant Web Services platform—and that obviously has some advantages. A single implementation project can give you a highly-reusable solution for sharing mainframe data, using industry-standard protocols and techniques.

On the downside, these projects are often extremely expensive because the vendors that implement them know that you're desperate and that you have few options. Because you're modifying the mainframe itself, this type of project will again require executive commitment, which may or may not be forthcoming. Project implementations are often lengthy, as you'll need to create maps and provide business logic between the Web Services layer and the underlying applications and data. In many ways, this approach shares the same disadvantages as the "top-down" approach, although this approach generally provides better cross-project reusability, so there's a better chance for capturing a good ROI in the long run.

There's also a real performance risk to the mainframe because you'll be asking it to perform a kind of work that many mainframe operating systems (OSs) weren't specifically designed for. You'll potentially be loading a great deal of additional transactional workload to the system, so you'll need to be very careful about estimating that traffic and its impact on the core of your business.

Overall, how does this approach stack up?

Criteria	Top-Down Approach	Modify the Mainframe App	Web Services on the Mainframe
Time to implement for a specific project	Lengthy	Moderate to Lengthy	Moderate to Lengthy
Expense	High	Moderate to High	High
Business risk	High	High	Moderate to High
Reusability	Very good, if well-designed	Poor	Can be very good
Mainframe performance impact	Moderate to high	Low to Moderate	Moderate to High

It's probably the best approach so far, but it's still expensive, and because it touches the mainframe directly, it's risky in terms of business impact and in terms of performance hit.

A Wish List for Cloud-Connected Legacy Data

Looking at these three approaches, can we construct a wish list that avoids some of their common disadvantages while perhaps including some of their advantages? Of course we can. We want an approach that

- can be implemented quickly, ideally for a single project.
- can be extended at any time to support additional projects—without having to worry about what those might be up-front.
- is less expensive than modifying the mainframe application directly.
- ideally, lives *off* of the mainframe. This setup minimizes or eliminates the need for executive-level commitment and reduces the chance for negative business and performance impact.

As you'll learn in the next article in this series, we can definitely meet these needs.