

Xbridge Customer Case Study: Kansas Department of Transportation

Kansas Department of Transportation increases user productivity by 20% while saving \$200,000 in the first year



SITUATION

The Kansas Department of Transportation (DOT) needed to get current information on the status of purchase orders that had been stored, for years, across three Virtual Storage Access Method (VSAM) tables and one DataBase 2 (DB2) table. The organization used a third-party data engine to extract the data, which was programmatically converted to XML. The Adobe® forms engine then consumed the XML and presented the information in an HTML-based Web form. Although the third-party data engine had been fairly costly, it could not support simultaneous requests from users and, far too often, would crash.

With these limitations, requests were often delayed (up to three days), dropped or simply became “runaways.” Users could not sort the data, which was “typically out of date and hard to use effectively.” A new request was required every time a Purchase Authority (PA) was needed. The enterprise data analytics could not process more than one request at a time.

The DOT, which was already using Xbridge Host Data Connect™ for other inquiries, preferred to keep their options open regarding Java or .NET. Therefore the Xbridge application had to only lightly impact their infrastructure. Additionally, the DOT wanted to replace the third-party data engine, keep the Adobe forms engine and produce the same XML for the forms engine to consume.

SOLUTION – HOST DATA CONNECT, XML & ASP.NET

Xbridge Host Data Connect enables the staff to get the information they need, on demand. Host Data Connect enables Windows® applications and browsers to securely access corporate data in

System/390® and z/OS™ mainframes, and to present that information in the format of the Adobe Form Server, all in real time. Host Data Connect supports both XML and ASP.NET, capabilities that are essential for the success of this application.

Here's how it works. The user calls up a form from the department's Adobe Form Server and enters information for a query.

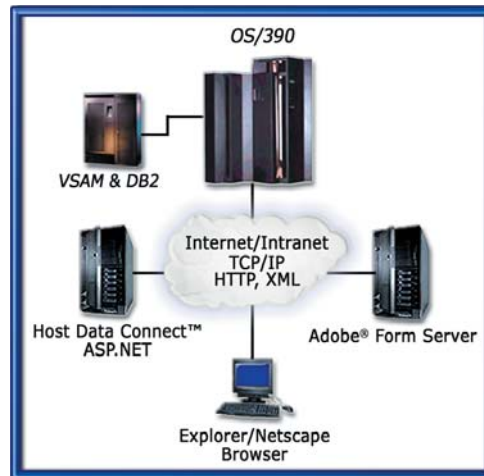
The query is passed onto Xbridge Host Data Connect, which accesses a DB2 table and several VSAM files on the department's mainframe.

A custom ASP.NET application leverages Xbridge technology to build an XML representation of the requested data. The Adobe forms engine then consumes the XML and presents the information in an HTML-based Web form, as it did in the previous application.

IMPLEMENTATION

Implementation of the Xbridge application was both challenging and straightforward.

The base Host Data Connect product provides a set of .NET components that can read data from the mainframe and produce XML. While the Xbridge out-of-the-box controls were not an exact fit, they provided a significant head start for the DOT customized application. The product reads one host table at a time, but the DOT data was stored across three VSAM tables and one DB2 table. Host Data Connect supports joins, but this was not a viable option since the VSAM data was stored in an awkward configuration. In one case the lines of the purchase order (PO) were stored in up to 99 different VSAM records with each record holding up to 15 lines. This unusual relationship made it difficult to directly use the Xbridge out-of-the box



controls. However, the controls could extract the data from the four sources. Because the controls were written in .NET, it was easy to write a few lines of code to customize an application that could combine these sources to generate a single XML output. Because the controls were written in .NET, it was easy to write a few lines of code to customize an application that could combine these sources to generate a single XML output.

XML MANIPULATION

To create the final XML output, Xbridge simply combined four components onto a Web page and configured them to access the four different data sources. Now, after a user specifies a PO number, the custom application creates four filters and directs the component to query the different data sources, storing the intermediate results in DataTables, a .NET-provided data structure. A fairly simple application was written to read each DataTable one record at a time, gathering the information needed to build the final XML output.

Simplifying the process, the .NET Framework Class Library (FCL) provides both a DOM-based Document class and an XMLWriter class. The DOM creates an in-memory model of the XML document and provides methods for writing the XML to a file or stream. The XMLWriter class only provides methods for writing XML to a file or stream. The XMLWriter is much more efficient, but is a forward-only mechanism, so Xbridge

used the Document class. The system programmatically reads records from the four Host Data Connect components (filtered on the PO number) and builds up an in-memory DOM representation. Once the DOM is complete, the Xbridge system outputs the DOM as XML to a string data stream that Xbridge then converts to the Web-page response object that is sent to the Adobe forms engine.

Had the DOT been implementing a new system from scratch, it probably would have made more sense to build a solution based on Web services. But the consuming application already existed and was not configured to consume a Web service. Therefore a custom application was more cost-effective since the consuming application needed virtually no modification.

It took less than two weeks to develop, test and implement the customized application. .NET was only installed on one existing Web server. The resulting system supports multiple simultaneous users and is far more reliable than the previous system. As a bonus, the system supports data analytics, a capability that wasn't available in the original application.sources.

XML DOCUMENTATION

As mentioned previously, the DOT had been debating whether to standardize on Java or .NET. Shortly after this project's successful implementation, the organization standardized on .NET. The Xbridge project may or may not have impacted that decision. However, since that decision, the DOT has successfully extended this application and created other applications. The DOT simply used the Host Data Connect control and the provided XML documentation, with little additional technical support from Xbridge.

The Kansas Department of Transportation ended up with a custom ASP.NET application that leverages Xbridge technology to build an XML representation of the requested data. The forms engine processes the XML-formatted data and presents it to the user in HTML.

RESULTS

Xbridge has been able to meet the department's needs for on-demand access to real-time data. The Xbridge implementation provides this capability at a fraction of the cost and time the department had been spending on the data analytics product, which did not meet the staff's needs.

The new source of XML data is far more reliable than the XML stream generated by enterprise data analytics product. Also, the new XML stream can handle multiple simultaneous requests, something that the department had been unable to do even after working with the previous product implementation for two years.

Furthermore, the new application is so flexible the department has been able to enhance the system's error-handling capabilities to accommodate invalid user input as well as old or missing mainframe data.

The staff plans to increase the storage on the mainframe to several years of data now that they have a reliable way to access that data. The mainframe presently holds several months of PR, PO and invoice data.

RETURN ON INVESTMENT

With limited custom development, Xbridge has been able to meet the department's needs for ongoing demand access to real-time host data. The Xbridge implementation provides this capability at a fraction of the cost and time the department had been spending on a data-analytics product, which did not meet the staff's needs.

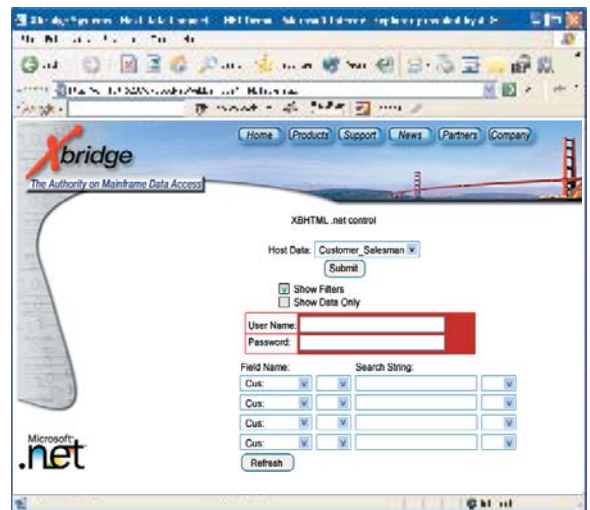
The Department of Transportation believes that, by enabling the staff to quickly and easily access real-time mainframe data, Host Data Connect has increased productivity

by 20%. The DOT estimates the product will save them \$200,000 in the first year by replacing the data-analytics package.

APPLYING .NET TO XBRIDGE

Xbridge Systems can quickly develop and implement these .NET applications partially because the company is so familiar with .NET, having incorporated support for the technology into its Host Data Connect product.

Xbridge has been developing applications to bridge the IBM mainframe and desktop worlds for years. Specifically, the company has developed powerful tools that allow Windows, Web and mobile applications to access data from VSAM, Queued Sequential Access Method (QSAM), DB2, Information Management System (IMS™) and Innovation Access Method (IAM®) databases located on OS3/90 or z/OS mainframes. These applications, which were developed using what Microsoft calls "classic" programming techniques, rely on OLE/DB standards and COM/DCOM technologies. Since the introduction of .NET, Xbridge has been gradually augmenting its tools with .NET applications. This gradual transition allows the company and its customers to adopt the .NET technology where it helps the most, and also maintain classic applications where appropriate.



Xbridge Systems, Inc.
1941 Landings Drive
Mountain View, CA 94043
(866) 356-1515
www.xbridgesystems.com