

Leveraging IBM i resources with LANSOpen for .NET

Once upon a time there was the IBM AS/400 and there was Microsoft Windows – two worlds and worlds apart. Combining data from these two worlds was a difficult undertaking, as it required knowledge of both. The situation gave rise to dual development teams, each team with different skills and priorities.

The world has moved on and today there are a number of products available that claim to narrow the gap between the IBM i and Windows platforms, such as 5250 emulators, screen scrapers and ODBC/JDBC data access tools.

Platform Divide Still a Challenge

The problem is that these products only narrow parts of the gap. After having the two platforms running side-by-side for over two decades, many IT managers are still challenged managing the platform divide. For example, how can you be assured that the same rules apply when a .NET program and an RPG program both update a database?

What we need is an effective mechanism for mixed development environments to share resources. We need a tool that bridges the gap completely.

Companies that need to develop with .NET tools and incorporate resources from the IBM i can do so with LANSOpen for .NET.

What is LANSOpen for .NET?

LANSOpen for .NET is a class library. The class library exposes services (including data and programs) that .NET developers can incorporate into Windows rich client and Web applications. The class library DLL is deployed with the .NET application.

Since 1992, LANSOpen has supplied 'LANSOpen Server' (a.k.a LANSOpen Open), a tool for building Windows applications that can access data and programs on the IBM i server. With LANSOpen Server, you installed components on the server and on the client

where the application executed. LANSOpen for .NET extends this capability with Web services. Using LANSOpen for .NET and C#.NET, you can write a Web service to act as a gateway to RPG, COBOL, Java and LANSOpen applications on the IBM i and the DB2 database. With this architecture, any application that can consume the Web service has access to resources on an IBM i server.

Using LANSOpen for .NET, a .NET programmer can:

- Write C# or VB.NET programs that access data and programs on the IBM i server from Visual Studio without having to know there is another server involved.
- Create and publish Web services from the IBM i server.

LANSOpen for .NET overcomes the challenge of bridging different architectures and simplifies the management of siloed development teams. Windows developers can write .NET applications and take advantage of resources on the IBM i server without leaving Visual Studio. The .NET applications do not need to implement a duplicate set of business rules. The rules are defined in the central LANSOpen Repository and deployed via Object Access Modules. Defining and applying business rules from a central point is



ARCHITECTS CORNER
by Richard Lancaster

particularly advantageous for Windows rich-client applications, as the rules can change without the need to recompile or redeploy the client components of the application.

This article provides a few scenarios where LANSOpen for .NET could bridge the platform gap. You can:

- Retrieve data from DB2 databases on the IBM i server.
- Perform create, update, insert, delete operations against DB2 databases, applying the same business rules as your IBM i applications would apply, without having to duplicate the rules in your .NET environment and without having to recompile .NET programs when the rules change.
- Invoke programs on the IBM i server, including LANSOpen functions and programs written in RPG, COBOL or Java.
- Access information in the LANSOpen Repository, such as formula fields, multilingual definitions, help text, file definitions and field definitions.

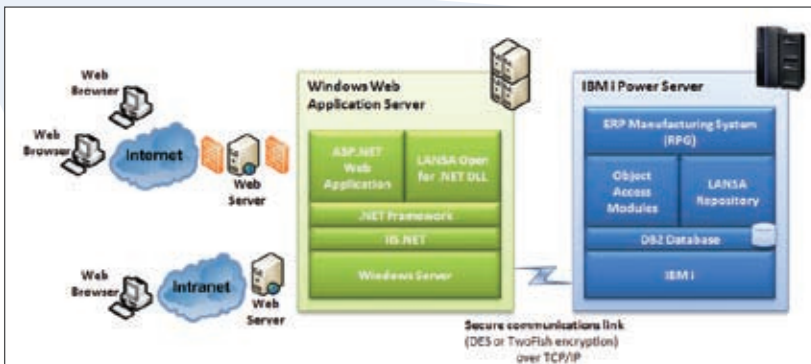
.NET Applications and LANSOpen for .NET

Scenario #1 Browser-based ASP.NET Web application on one side, RPG application on an IBM i server on the other side

Suppose a company wants to improve customer service and reduce the volume of inquiry calls by providing customers with Web self-service access to their orders. The manufacturer's ERP package on the IBM i provides a 5250 screen for order enquiry that is inappropriate for the Web context.

As the company's policy is to deliver their Web sites via Windows servers, they decide to build an ASP.NET Web application that will extract information from the ERP system on the IBM i, using LANSOpen for .NET.

Because the ERP system is not a LANSOpen application, the database definitions associated with orders have to be 'made known' to the LANSOpen Repository. This process is automated. In addition, formula fields, validations and other business rules can be added to the LANSOpen Repository without affecting the ERP



Scenario #1 — Browser-based ASP.NET Web application on one side, RPG ERP application on an IBM i server on the other side.

system itself. The LANS Open for .NET DLL will publish the table definitions for use by the ASP.NET Web application.

The .NET developer only has to build the ASP.NET Web application and pass calls to the LANS Open for .NET DLL. The .NET developer does not need to know that the ERP system resides on the IBM i server

Suppose the company wants to give customers the option to update their ship-to details. The optimum approach is to define the validation rules in the LANS Repository. LANS's Object Access Modules (OAMs) apply the validation rules and return any error messages, which LANS Open for .NET passes back to the ASP.NET application. The ASP.NET Web application only needs minor change and an exception handler to display possible error messages. When the update request passes all validations, the OAMs will update the database.

The validation rules may be changed without having to recompile or redeploy the ASP.NET Web application.

Scenario #2 Rich-client .NET application on one side, LANS-based application and IBM i database server on the other side

Suppose a company wants to grant its sales people update access to the customer details. The Customer Management System (CMS) resides on the IBM i and is written in LANS, so the business rules governing the CMS already reside in the LANS Repository.

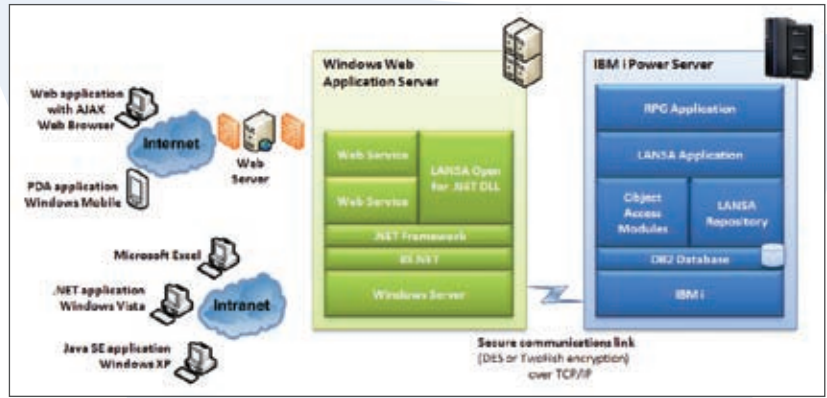
The company decides to build a rich-client .NET application written in C# to deploy to the sales people's laptops and notebooks. Through LANS Open for .NET, the .NET developer will see the CMS as if it were another .NET application. LANS OAMs will enforce the business rules and any errors will be returned to the .NET application via LANS Open for .NET.

This solution reuses the existing CMS, hides the architecture differences and ensures data integrity.

Web Services and LANS Open for .NET

The Web services scenario is a bit more complex than the previous scenarios, but it is flexible and wide ranging in its application potential.

The Web service is a C#.NET program which uses LANS Open for .NET to access data and programs on the IBM i server. For example, the Web service can provide customer data from a DB2 database to a Windows-based Web application. The Web application will call the Web service to request



Scenario #3 and Scenario #4 — Web services published from LANS and RPG applications on an IBM i server.

customer details. The Web service will invoke LANS Open for .NET to retrieve the data and then return the customer details to the Web application.

The Web service will be a simple program that acts as a mediator to manage the requests and responses. LANS Open for .NET, the OAMs and the LANS application do the real work by applying the business rules and retrieving (or updating) the DB2 data.

This architecture applies to situations where you need to incorporate data or other IBM i resources into an application. Typically, the application consuming the Web service will manage the user interface and will use the Web services to supply data or to initiate an activity on a remote system.

LANS Open for .NET simplifies the transformation of resources on an IBM i server into Web services.

Scenario #3 Microsoft CRM needs to publish customer address changes to multiple line-of-business systems on the IBM i

The company in this example uses a CRM system to provide a consolidated view and central entry point for customer information. The information is distributed and stored across several IBM i applications that support various business functions. Updating the customer address in the central CRM makes sense. The problem is how to publish the new address to each of the applications.

One approach is to create an address management function with LANS to validate address information and update the systems where the addresses are used. The Web services are built as .NET components. LANS Open for .NET will provide the interface to the address management function for the Web services and hide the complexity of address validation and address distribution from the .NET developer.

Scenario #4 Microsoft Excel spreadsheet populates itself with data from an IBM i-based financial system

Suppose the accountants of a financial management group want to analyze the debtors' ledger, residing on the IBM i. Being accountants they prefer Microsoft Excel as their working environment. The accountants need to be able to retrieve debtors' data.

One approach is to build a Web service to retrieve the debtors' data and an Excel application that will manage the selection criteria and load the data into a worksheet. The Web service will be a .NET application that accepts query parameters, interacts with LANS Open for .NET to access the debtors' data and returns a set of data to the Excel application.

The work for the .NET programmer is to build the Web service and the Excel application. LANS Open for .NET hides the source of the data from the .NET developer.

How Does LANS Open for .NET Help Manage Your Architecture Challenges?

LANS Open for .NET hides the gap between the .NET and IBM i platforms. The .NET developer sees only a set of classes in Visual Studio and does not need to know the nature of the server. From the developer's perspective, the gap is hidden completely.

LANS Open for .NET insulates the .NET developer from having to understand the IBM i. The developer builds the .NET application in Visual Studio and sees the data and processes from the IBM i server as resources in the .NET environment.

The scenarios described in this article provide only a glimpse of the possibilities for publishing data and processes from the IBM i world to the .NET world with LANS Open for .NET. ■