



Connecting SharePoint Server 2010 to Line-of-Business Systems to Deliver Business-Critical Solutions

White Paper
December 2011

DISCLAIMER

The information contained in this document represents the current plans of Microsoft Corporation on the issues presented at the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Schedules and features contained in this document are subject to change.

Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the expressed written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give any license or rights to these patents, trademarks, copyrights, or other intellectual property.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, the Microsoft logo, InfoPath, SharePoint Server, SQL Server, Visual Studio, and other product names are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.

Table of Contents

Who Should Read This White Paper?	5
The Case for Connectivity: Increased ROI and Decreased Business Risk.....	5
Connecting LOB Applications to SharePoint Server 2010.....	6
Exploring the Tools	6
<i>Business Connectivity Services</i>	6
<i>Enterprise Search</i>	8
Analyzing the Situation	8
Manufacturing Example: Fabrikam, Inc.	9
Identifying the Business Need	9
Connecting Systems with Business Connectivity Services.....	10
<i>Choosing an Integration Method</i>	10
<i>Building the Custom Connector</i>	11
<i>Deploying the Connector</i>	12
<i>Viewing an External Table as a SharePoint Server List</i>	13
<i>Using a List to Write Data Back to the LOB System</i>	16
Extending the Reach of LOB Data	17
Conclusion	18
Resources	19

Who Should Read This White Paper?

Business users at **every level of your organization** should have access to important data and be connected to processes that enable them to support operations. However, critical business data often is stored in disparate systems, and ad hoc processes block efficiencies. Users frequently call on IT to assist them in reaching and reconciling this business data, which can divert important IT resources away from strategic work that positions IT as a business partner, rather than a cost center.

This white paper is intended for enterprise architects who have the challenge of providing organizations and end users with an easier way to access the information locked in back-end systems. Using the interoperability capabilities of Microsoft® SharePoint Server® Server 2010—including Business Connectivity Services—IT can build cost-effective, manageable integrations to deliver increased productivity and return on investment (ROI) to their business clients.

This white paper:

- Explains why connecting line-of-business (LOB) applications to SharePoint Server makes business and technological sense.
- Discusses how an enterprise architect might design an integration between SharePoint Server and a back-end system.
- Highlights the technologies and techniques involved in building a sample integration.
- Explores additional options for working with enterprise data once it is exposed in SharePoint Server.

The Case for Connectivity: Increased ROI and Decreased Business Risk

Organizations have long relied on IT to “connect the dots” among disparate legacy systems. These points of connection traditionally have been ad hoc, delivering limited value but consuming many IT resources in the creation and deployment process. Using SharePoint Server 2010 to surface business data from LOB systems and build solutions can increase the ROI of legacy systems, speed solutions’ time-to-market, and connect users from multiple business functions through a single familiar interface—all of which free IT resources to focus on more strategic initiatives.

Business units can reduce training costs because SharePoint Server offers the familiar Microsoft Office experience, enabling people to quickly and easily adopt SharePoint Server versus receiving training on a variety of more complex LOB applications.

In addition, SharePoint Server can speed the time-to-market of otherwise time-consuming and resource-intensive solutions by taking advantage of native application development capabilities. Plus, powerful Search and BI capabilities provide wider end-user functionality, which can boost productivity, reduce costs, and increase user satisfaction.

All employees have the responsibility to make the best decisions possible, based upon the data available to them at any time. If their ability to analyze this data and transform it into useful information is improved, the overall quality of their decisions can be improved as well.

- Michael Schiff

Founder and Principal Analyst, MAS Strategies

Finally, SharePoint Server can help to reduce business risk by increasing the visibility of critical data. The ability to access accurate, real-time business data can have a major impact across an organization. In his 2009 white paper, "Business Intelligence: A Guide for Midsize Companies," MAS Strategies' Founder and Principal Analyst Michael Schiff said the following:

"All employees have the responsibility to make the best decisions possible, based upon the data available to them at that time. If their ability to analyze this data and transform it into useful information is improved, the overall quality of their decisions can be improved as well."

In other words, when you surface relevant data to the people who need it, when they need it, you enable them to make better decisions faster. This can reduce mistakes that result from misinformation and decrease overall business risk.

SharePoint Server also can reduce risk by enhancing security, privacy, and compliance through a flexible authentication model. This authentication model can help an organization to maximize its SharePoint Server 2010 deployment while maintaining highly secure control over corporate assets and increasing compliance.

Connecting LOB Applications to SharePoint Server 2010

A successful solution integration starts with solid planning. This section provides an overview of the key tools for such an integration, as well as a four-step framework for determining the right connectivity approach for each unique situation.

Exploring the Tools

As a productivity platform, SharePoint Server 2010 offers a range of options for creating solutions that connect users to business-critical information, whether it resides in SharePoint Server, LOB systems, or other unstructured repositories. This subsection highlights a few connectivity options that are especially applicable when building business-critical applications.

Business Connectivity Services

A primary interoperability component of SharePoint Server 2010, Business Connectivity Services (BCS) is responsible for reading from and writing to external systems. BCS in SharePoint Server enables connectivity to external data sources, such as databases and LOB systems (Figure 1).

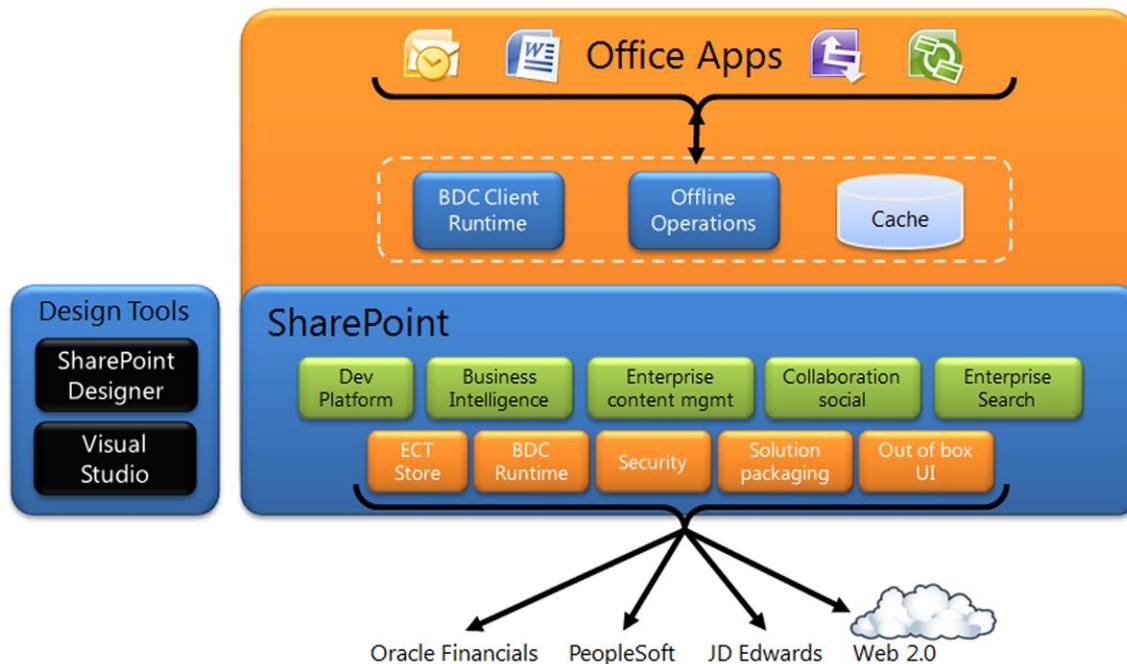


Figure 1: BCS architecture diagram

The Business Data Catalog (BDC) is the foundation of Business Connectivity Services, and the foundation of the BDC is the entity model that defines entities, methods, relationships, and so on. After these are defined, the BDC runtime object model is populated with data, and it then provides a uniform interface to the LOB data. Other base-level components of BCS support and extend the BDC with items like security and user interfaces.

Two key building blocks of SharePoint Server are **lists** and **content types**. A list, basically, is a collection of items that generally shares a common set of properties. Think of a list somewhat like a database table, with columns representing properties of items and rows representing the items themselves. A content type is a collection of metadata about an item (principally columnar properties, but also other behaviors) that can be reused across lists. Large parts of SharePoint Server functionality, including the widely used document libraries, are built on the foundation created by lists and content types.

Usually, a list contains items of one or more content types that have been created directly in SharePoint Server. However, the functionality offered by the list infrastructure is available to LOB data, as well. Using one of the most powerful capabilities of BCS, you can create *external* content types and *external* lists that present and interact with data in virtually any system or repository.

With the BCS set of components, external lists can provide full create, read, update, and delete (CRUD) capabilities to data stored in any system that can be connected to SharePoint Server. To users, an external list looks and feels almost exactly like any other SharePoint Server list: They can create new views, easily sort and filter data, and even take data offline using Microsoft SharePoint Server Workspace 2010. Moreover, site owners and other power users can create custom forms using Microsoft SharePoint Server Designer 2010 and Microsoft InfoPath® 2010; add list web parts to other pages; and create composite applications—all using LOB data. BCS uses a built-in or custom connector to ensure that this data is

properly synchronized with the back-end system. (**Note:** Building this connector is discussed in a [later section](#) of this white paper.)

Enterprise Search

Either in concert with Business Connectivity Services or as an independent component, the enterprise search capabilities offered by Microsoft SharePoint Server 2010 and Microsoft FAST Search Server 2010 for SharePoint Server can provide another way to expose and unlock data in LOB systems.

SharePoint Server 2010 Search provides an interactive, visual search experience. Visual cues can help users to find information quickly, while refiners let them drill into the results and discover insights, often in easier and more actionable ways.

Example: An account manager receives a request to adjust a custom order. Before responding to the request, she must determine whether any of her organization's warehouses have the items in stock to amend the order. Because the account manager's enterprise resource planning (ERP) system is connected to SharePoint Server 2010, she can simply open her team portal and search for the part, and she finds that it is available in two warehouses.

Analyzing the Situation

As an enterprise architect, you should follow four main steps when designing an integration solution that connects an LOB system to SharePoint Server:

1. **Identify the business process that needs improvement:** Be prescriptive about what you are trying to accomplish. Stakeholders should articulate and agree on a compelling business need that underpins the desire to surface LOB data in SharePoint Server. For example, the data might be linked to a core business process.
2. **Establish which LOB application (or applications) you are targeting:** The application targeted will help to determine the most appropriate way to connect. In addition to its own data model and interoperability strategy, each LOB system has a particular philosophy of working with the information that its users provide and consume. To some degree, these details will steer your design. For example, if a certain system uses a data model that is similar to the SharePoint Server tabular structure, it might make sense to connect at the data tier. Conversely, a system that provides much value in its unique presentation of data might be better connected at a higher level.
3. **Determine what data is appropriate to bring into SharePoint Server:** This step should be directly related to the business need being addressed and the process requiring improvement. By targeting specific data, you can better avoid data redundancy and maintain security. In short, surface only the data you need.
4. **Decide how you want users to interact with the data once it is in SharePoint Server:** By evangelizing the solution, you can help to speed the adoption process and see faster ROI. Provide prescriptive guidance that details how the solution can improve job functions and what, specifically, this means to users in terms of how they interact with systems. Consider communicating this guidance in the context of a before-and-after scenario, if appropriate.

These four steps can help you to define your approach to connecting LOB applications to SharePoint Server. Depending on your business goals (as stated above in [The Case for Connectivity](#)), the approaches

you take will differ. The next section applies this situational analysis in the context of a fictional business setting. The example discusses connecting the JD Edwards EnterpriseOne ERP system to SharePoint Server. Remember, however, that JD Edwards is just one of many systems that can be connected to SharePoint Server.

Manufacturing Example: Fabrikam, Inc.

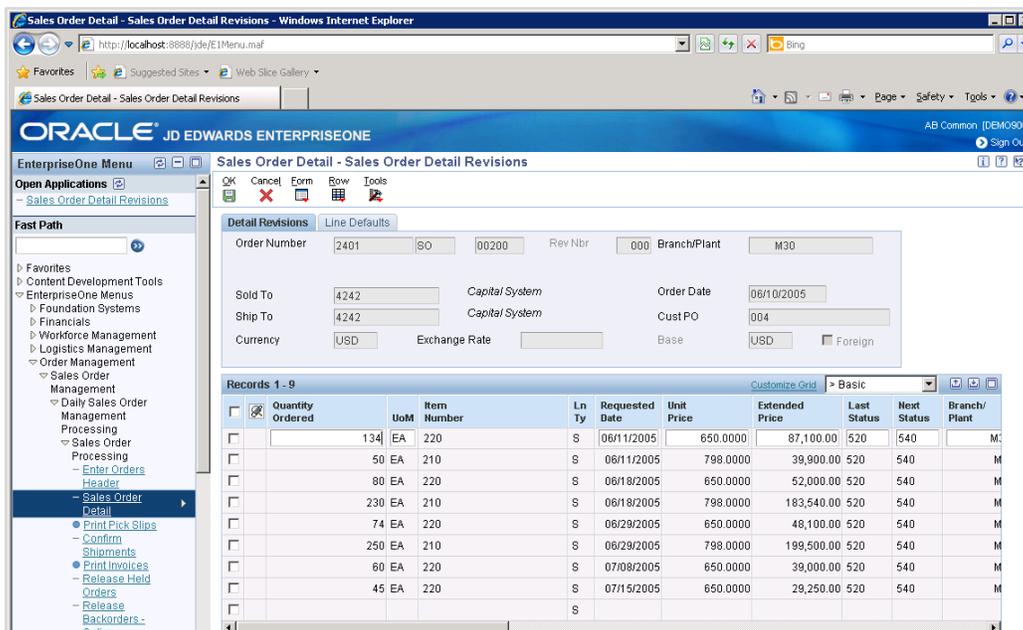
Fabrikam, Inc., is a made-to-order manufacturing organization that operates several disparate systems across its business functions. In addition to various departmental implementations of small- and medium-sized systems, the company uses JD Edwards EnterpriseOne as its ERP suite for manufacturing. Fabrikam also has installed SharePoint Server 2010, which is rapidly gaining popularity as a collaboration platform.

Identifying the Business Need

Recently at Fabrikam, a product line has become so popular that customers are frequently increasing their standing orders. Therefore, LOB managers have identified the order change process as one that can be improved by connecting their JD Edwards EnterpriseOne ERP system to SharePoint Server. The managers want to enable direct communication between salespeople and the plant floor, with the goal of preventing requests for additional products after an order has entered production. It is a business-critical need for manufacturers like Fabrikam to improve production processes and avoid short-run orders when possible.

Currently, the process for communicating and initiating an order change happens in three manual steps:

- Salesperson sends an email message to the Planning and Scheduling Department.
- Department worker enters the order change into the JD Edwards system (Figure 2).
- Department worker sends an email message to the production team to ensure that the change is reflected in the manufacturing execution system.



The screenshot displays the Oracle JD Edwards EnterpriseOne interface. The main window is titled "Sales Order Detail - Sales Order Detail Revisions". It shows a form with the following fields:

- Order Number: 2401
- SO: 00200
- Rev Nbr: 000
- Branch/Plant: M30
- Sold To: 4242 Capital System
- Order Date: 06/10/2005
- Ship To: 4242 Capital System
- Cust PO: 004
- Currency: USD
- Exchange Rate: [empty]
- Base: USD
- Foreign:

Below the form is a table titled "Records 1 - 9". The table has the following columns: Quantity Ordered, UoM, Item Number, Ln Ty, Requested Date, Unit Price, Extended Price, Last Status, Next Status, and Branch/Plant. The data rows are as follows:

Quantity Ordered	UoM	Item Number	Ln Ty	Requested Date	Unit Price	Extended Price	Last Status	Next Status	Branch/Plant
134	EA	220	S	06/11/2005	650.0000	87,100.00	520	540	M
50	EA	210	S	06/11/2005	798.0000	39,900.00	520	540	M
80	EA	220	S	06/18/2005	650.0000	52,000.00	520	540	M
230	EA	210	S	06/18/2005	798.0000	183,540.00	520	540	M
74	EA	220	S	06/29/2005	650.0000	48,100.00	520	540	M
250	EA	210	S	06/29/2005	798.0000	199,500.00	520	540	M
60	EA	220	S	07/08/2005	650.0000	39,000.00	520	540	M
45	EA	220	S	07/15/2005	650.0000	29,250.00	520	540	M

Figure 2: Sales orders displayed in JD Edwards EnterpriseOne

Unfortunately, the sales team's emails to the Planning and Scheduling Department often get buried, and changes are not entered before the original order goes into production. This results in additional time, effort, and resources required to correct the order. To streamline the process and use resources more effectively, the LOB managers have requested the sales team to enter order changes through SharePoint Server and IT to connect the two systems.

Connecting Systems with Business Connectivity Services

As an enterprise architect, you need to help Fabrikam simplify the order change process by connecting the JD Edwards ERP system to SharePoint Server 2010. By using the SharePoint Server platform, you can help end users to quickly and easily access the data they need from a familiar interface.

You analyze the business requirements and find that users need to be presented with custom views of sales data so that they can share up-to-date information and take appropriate action. You decide that BCS provides the best approach to completing the integration. With this choice made, you begin to work with developers and other IT professionals to design and build the BCS connection to the JD Edwards system.

Choosing an Integration Method

BCS supports multiple methods for connecting to external systems:

- **SQL Server:** BCS can access data stored in Microsoft SQL Server® databases without requiring a developer to write custom code. SharePoint Server Designer 2010 is used to configure the mappings (or model) that will expose SQL Server-hosted data as external lists and content types.
- **Other Databases:** BCS natively connects to other relational databases, including Oracle, OLE DB, and ODBC connections. When connecting to non-SQL Server databases, the model must be created in an XML-based file and imported into SharePoint Server. Alternatively, the code-based options discussed below can be used to connect to the database and interface with BCS.
- **Web Services:** BCS easily consumes Windows Communication Foundation (WCF) services. Again, using SharePoint Server Designer 2010, BCS can be configured to map list operations to WCF service methods. When interacting with LOB systems that may not expose WCF-compatible services, a developer can write a custom middle-tier service to connect natively to the back-end system and expose the necessary functionality as WCF methods.
- **.NET Connectors:** Using Microsoft Visual Studio® 2010 and the included SharePoint Server project templates, a developer can create connectors that employ the full flexibility of the .NET Framework to interact with LOB systems using standard and established integration patterns. By implementing certain interfaces, the patterns expose the necessary functionality for BCS to work with these connectors.

To a large degree, the choice of method depends on the capabilities of the specific LOB system to which you want to connect. If the system uses SQL Server or exposes applicable WCF services, the built-in, no-code-required connectors often are the most cost-effective and simplest choices.

In this case, JD Edwards EnterpriseOne is your target. A complex ERP system with a platform all its own, EnterpriseOne exposes a wide variety of interoperability options, including a Business Services framework where units of functionality can be encapsulated and exposed as web services and several database-tier

integration points. (A full exploration of EnterpriseOne interoperability and best practices is beyond the scope of this paper).

Each implementation of each enterprise system is unique. The .NET connector approach to creating a BCS integration gives you the flexibility to interact with virtually any of these interoperability technologies. In the Fabrikam example, after consulting with the manufacturing IT team that runs the EnterpriseOne system, you decide to create a .NET connector to interact with the sales order tables in the system's Oracle Database 11g database.

Building the Custom Connector

You have analyzed the business problem and chosen your approach. Now it is time to engage the development team to build the connector for BCS and the JD Edwards system.

Using Visual Studio 2010, the team creates a standard C# Class Library project to contain the connector's code. The developers reference two assemblies that ship with SharePoint Server 2010:

Microsoft.BusinessData and **Microsoft.SharePoint Server** (Figure 3). These assemblies contain the APIs that are necessary for implementing BCS connectors.

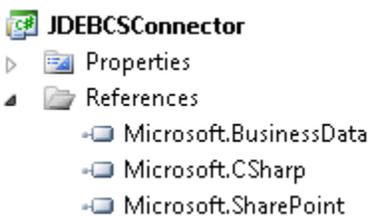


Figure 3: Assemblies with APIs for BCS connectors

In addition, the development team references and configures the Oracle Data Provider for .NET to allow connection to Fabrikam's EnterpriseOne back-end system.

Several interfaces are provided for creating code related to BCS: **ISystemUtility** is the main interface used to read and write data from the external system. For the Fabrikam team, the bulk of the development work is in implementing the **ExecuteStatic** method of this interface. ExecuteStatic is responsible for gathering input data from the BCS framework and executing operations against the EnterpriseOne database to perform the requested action.

Several types of operations can be conducted within the ExecuteStatic method (Figure 4). For Fabrikam, you need instances of the following method types:

- **Finder:** A finder returns a list of instances of an entity (in this case, sales orders). The Finder method accepts, optionally, a number of filter parameters that determine which items are returned. Examples of filter parameters include company name and branch/plant location.
- **SpecificFinder:** The SpecificFinder returns a single entity (in this case, one sales order). If necessary, you can have the SpecificFinder return a different (perhaps larger) set of properties than the Finder.
- **Creator:** This method type is used to add a new sales order to the system.
- **Updater:** Similar to the Creator, this update method type is responsible for making changes to an existing sales order.

```

public void ExecuteStatic(Microsoft.BusinessData.MetadataModel.IMethodInstance mi,
    Microsoft.BusinessData.MetadataModel.IJobSystemInstance si,
    object[] args,
    IExecutionContext context)
{
    if(mi.MethodInstanceType == Microsoft.BusinessData.MetadataModel.MethodInstanceType.Finder)
    {
        // Get Sales Orders based on filter values
    }
}

```

Figure 4: Operations within the ExecuteStatic method

The Microsoft.BusinessData namespace also contains other interfaces that can be added to your connector to increase flexibility and robustness. For example, developers can choose to override the BCS default connection management functionality and implement a custom connection manager that is more appropriate for a particular environment (**IConnectionManager**), or they can provide parameters for administrators to configure the connector (**IAdministrableSystem**).

Deploying the Connector

After the Fabrikam development team builds the connector, you need to deploy it to your SharePoint Server farm. The custom connector assembly must be placed in the Global Assembly Cache of the web front-end and application servers of the farm. The SharePoint Server solution deployment functionality can be used to create a WSP package for deploying the assembly across the farm.

In addition to the assembly, you need a **model**, or .bdcm file. The model is an XML file that tells BCS how to interact with a custom connector. The model file contains configuration information about the connector assembly and each of the entity types (in this case, sales orders).

With the connector assembly installed and the model created, you now can deploy the connector to SharePoint Server. Using the SharePoint Server Central Administration web-based user interface, you can import the model XML to the BDC service (Figure 5).

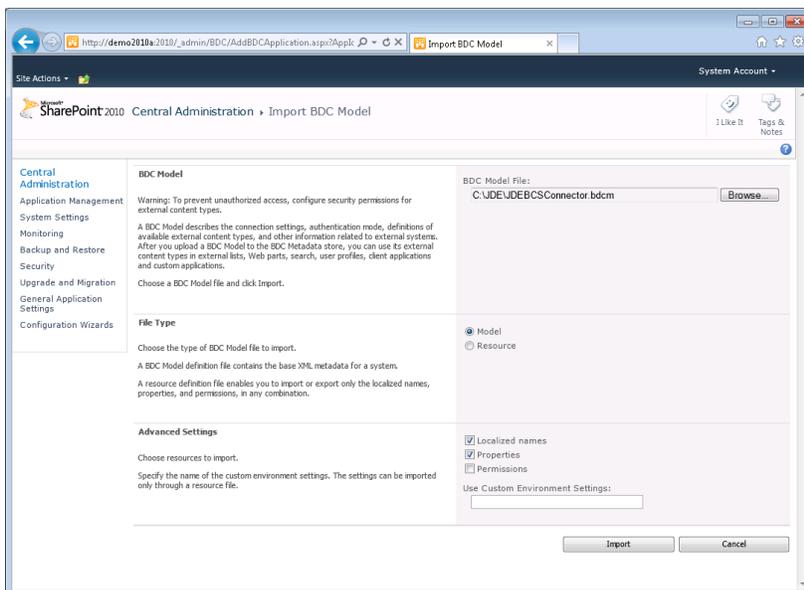


Figure 5: Importing a model

The BDC service validates the model, ensuring that there are no discernible errors in the mappings and configurations. It then imports the model. The model and its associated connector are now available as a data source for SharePoint Server lists (Figure 6).

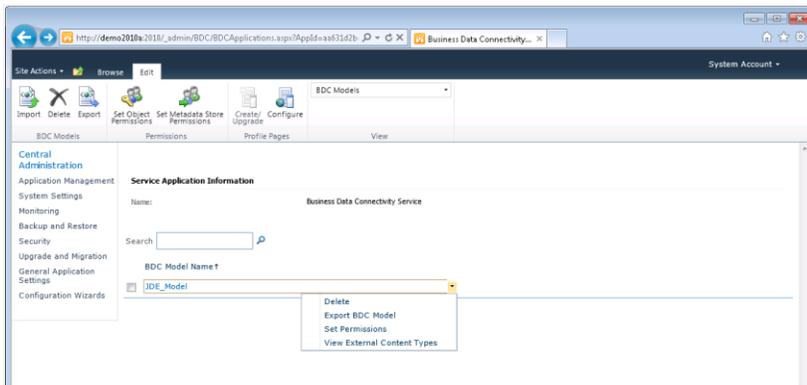


Figure 6: Imported BDC model

Viewing an External Table as a SharePoint Server List

With deployment complete, the connector work is finished. Now it is time to bring the newly exposed sales order data into one of the sales team’s SharePoint Server sites. You navigate to the Western Region sales team’s existing site and, using the SharePoint Server web-based user interface, create a new list (Figure 7).

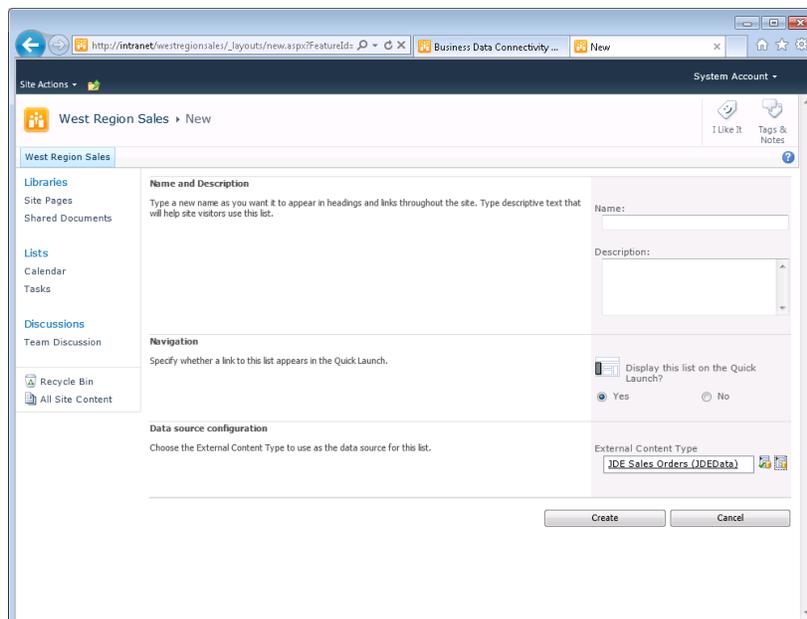
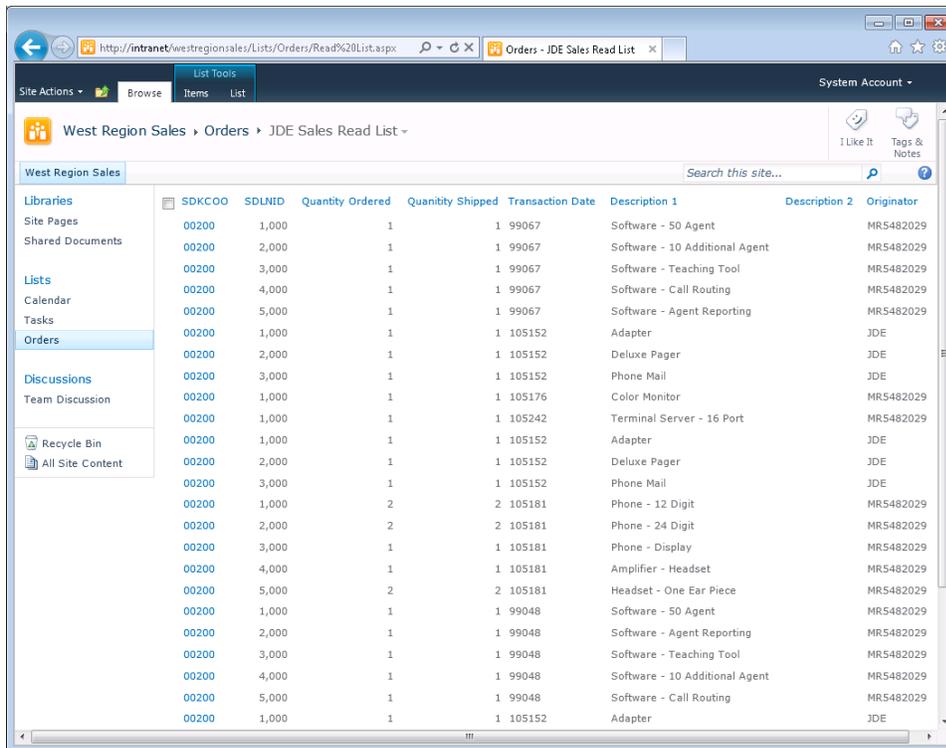


Figure 7: Creating an external list

Using the External List template, you have the option to choose the new external content type that was created when you deployed the model to BCS. Name the list “Orders” and click OK. You are taken to the list’s default view, where you can see a list of orders from EnterpriseOne (Figure 8).



SDKCOO	SDLNID	Quantity Ordered	Quantity Shipped	Transaction Date	Description 1	Description 2	Originator
00200	1,000	1	1	1 99067	Software - 50 Agent		MR5482029
00200	2,000	1	1	1 99067	Software - 10 Additional Agent		MR5482029
00200	3,000	1	1	1 99067	Software - Teaching Tool		MR5482029
00200	4,000	1	1	1 99067	Software - Call Routing		MR5482029
00200	5,000	1	1	1 99067	Software - Agent Reporting		MR5482029
00200	1,000	1	1	1 105152	Adapter		JDE
00200	2,000	1	1	1 105152	Deluxe Pager		JDE
00200	3,000	1	1	1 105152	Phone Mail		JDE
00200	1,000	1	1	1 105176	Color Monitor		MR5482029
00200	1,000	1	1	1 105242	Terminal Server - 16 Port		MR5482029
00200	1,000	1	1	1 105152	Adapter		JDE
00200	2,000	1	1	1 105152	Deluxe Pager		JDE
00200	3,000	1	1	1 105152	Phone Mail		JDE
00200	1,000	2	2	2 105181	Phone - 12 Digit		MR5482029
00200	2,000	2	2	2 105181	Phone - 24 Digit		MR5482029
00200	3,000	1	1	1 105181	Phone - Display		MR5482029
00200	4,000	1	1	1 105181	Amplifier - Headset		MR5482029
00200	5,000	2	2	2 105181	Headset - One Ear Piece		MR5482029
00200	1,000	1	1	1 99048	Software - 50 Agent		MR5482029
00200	2,000	1	1	1 99048	Software - Agent Reporting		MR5482029
00200	3,000	1	1	1 99048	Software - Teaching Tool		MR5482029
00200	4,000	1	1	1 99048	Software - 10 Additional Agent		MR5482029
00200	5,000	1	1	1 99048	Software - Call Routing		MR5482029
00200	1,000	1	1	1 105152	Adapter		JDE

Figure 8: The sales order list from JD Edwards EnterpriseOne

Notice that this list shows an extremely limited number of parameters about each item as compared to the full EnterpriseOne view. This is by design: You have limited your model to only those fields relevant to the process you want to improve. Before releasing this new list to the salespeople of the Western Region, it is important to be sure that you are exposing the correct and most relevant data for their process.

To achieve this, edit the default view of the sales order list. The data source filters that were configured as part of your model can be set up here. Specify the location code that represents the Western Region (Figure 9). In addition, you can use SharePoint Server filtering capabilities to further refine the list. Note, however, that it is often far more efficient to design the filter logic into the LOB connector rather than relying on the presentation-layer technology to do this filtering.

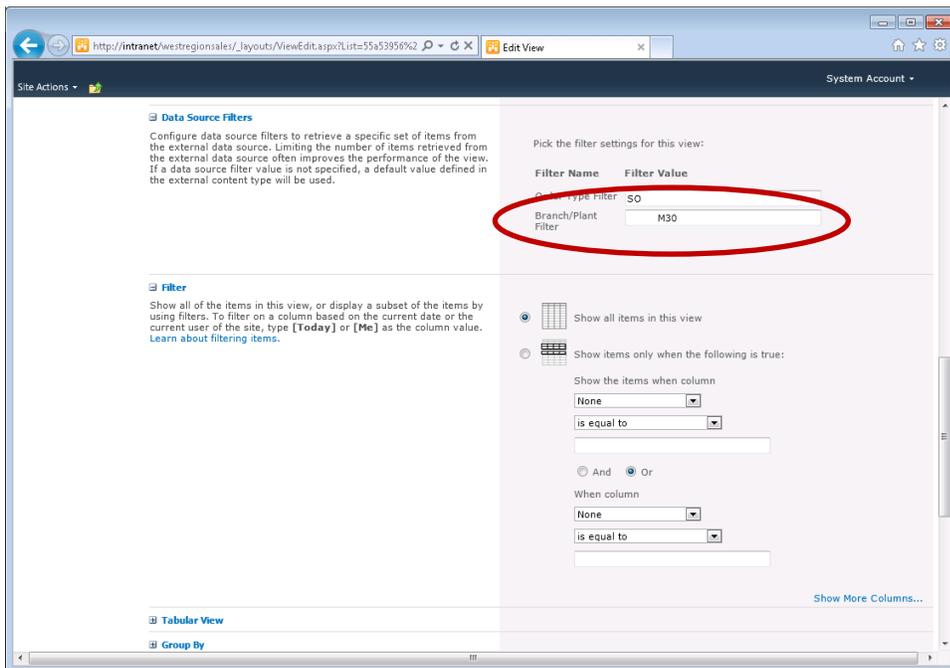


Figure 9: Filtering the list, with the "Branch/Plant" location code highlighted

Now, Fabrikam salespeople can browse to this list and view their upcoming orders (Figure 10).

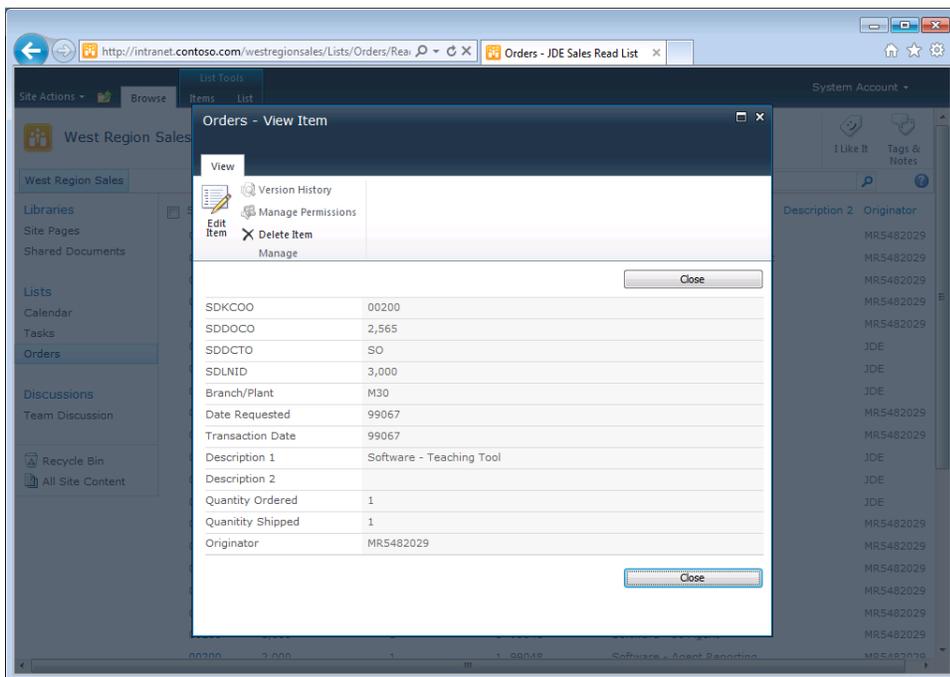
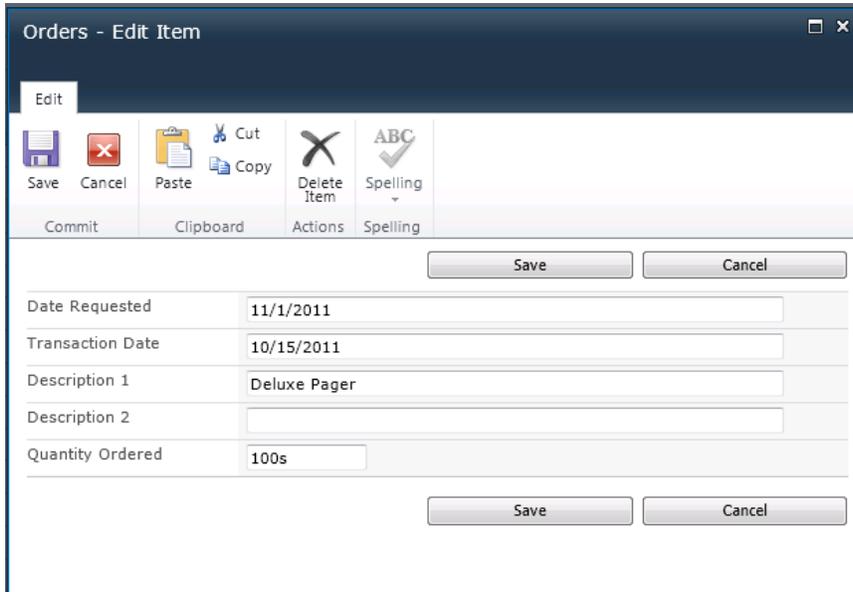


Figure 10: List item view

Using a List to Write Data Back to the LOB System

Remember that the previous [Building the Custom Connector](#) subsection discussed the Creator and Updater methods for implementation. Because these methods encapsulate all of the complex business logic necessary to insert and update items in JD Edward EnterpriseOne, you can simply use the SharePoint Server built-in list editing capability to allow the limited updates needed to improve Fabrikam's order change process (Figures 11 and 12).

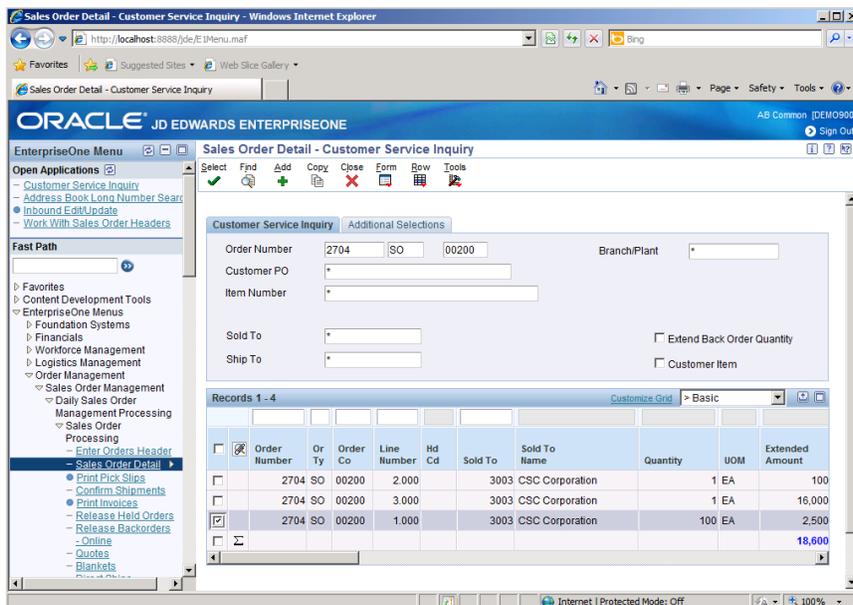


The screenshot shows a dialog box titled "Orders - Edit Item" with a standard Windows-style title bar. Below the title bar is a ribbon with an "Edit" tab. The ribbon contains several groups of icons: "Commit" (Save, Cancel), "Clipboard" (Paste, Copy), "Actions" (Delete Item), and "Spelling" (Spelling). Below the ribbon are two "Save" and "Cancel" buttons. The main area of the dialog contains several text input fields:

- Date Requested: 11/1/2011
- Transaction Date: 10/15/2011
- Description 1: Deluxe Pager
- Description 2: (empty)
- Quantity Ordered: 100s

At the bottom of the dialog, there are two "Save" and "Cancel" buttons.

Figure 11: Editing a sales order



The screenshot shows the Oracle JD Edwards EnterpriseOne interface. The browser address bar shows "http://localhost:8888/jde/EMenu.maf". The main window title is "Sales Order Detail - Customer Service Inquiry". The interface includes a menu on the left, a toolbar, and a main data entry area. The "Customer Service Inquiry" form shows the following fields:

- Order Number: 2704 SO 00200
- Customer PO: (empty)
- Item Number: (empty)
- Sold To: (empty)
- Ship To: (empty)

Below the form is a table with the following data:

Order Number	Or Ty	Order Co	Line Number	Hd Cd	Sold To	Sold To Name	Quantity	UOM	Extended Amount
2704	SO	00200	2.000		3003	CSC Corporation	1	EA	100
2704	SO	00200	3.000		3003	CSC Corporation	1	EA	16,000
2704	SO	00200	1.000		3003	CSC Corporation	100	EA	2,500
									18,600

Figure 12: Updated sales order in EnterpriseOne

In addition, new items can be added using the same pattern. A salesperson can simply select **New Item** from the SharePoint Server Ribbon and enter the necessary data (Figure 13).

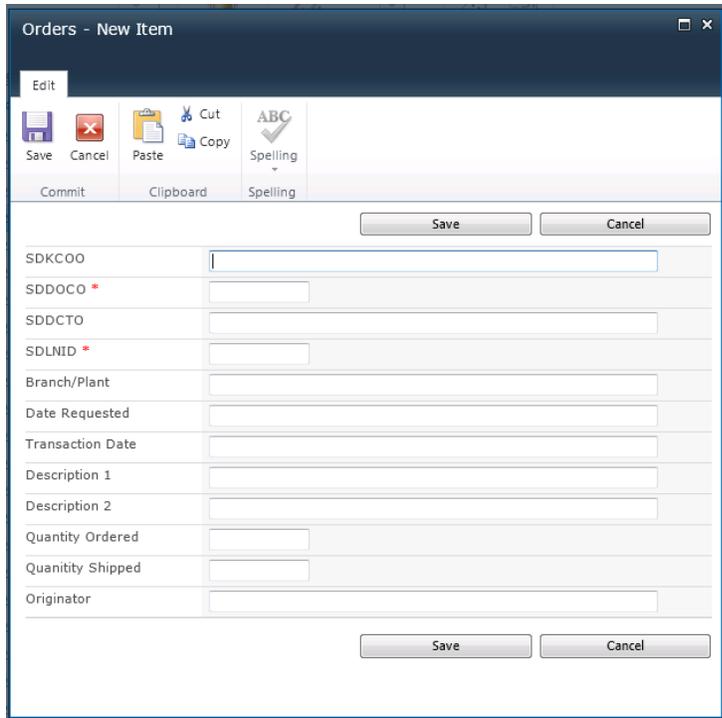


Figure 13: Adding a sales order

Extending the Reach of LOB Data

Using the SharePoint Server list interface, you have fulfilled Fabrikam’s business goal of streamlining the sales order change process. This improvement eliminates manual communication steps and, ultimately, avoids costly additional production runs. However, this is not the limit to how your organization can use this business-critical data. Once the data is in SharePoint Server 2010, you have the flexibility to use it for other purposes, including:

- **Implement, Extend, and Improve Business Processes:** Find and visualize the information you need in SharePoint Server 2010. Take advantage of out-of-the-box platform capabilities like collaboration, social computing, and content management to enable the right people to access the right information at the right time. IT can design and administer solutions quickly so that users can build their own templates and workflows to connect business data to their processes.
- **Gain Additional Productivity:** SharePoint Server provides several capabilities, including Search and Insights, that can help organizations to improve workforce productivity and visualize business data in real-time. These capabilities have built-in security and manageability to help ensure safe and easy use.
- **Define and Measure Success:** SharePoint Server 2010 Insights provides interactive dashboards and scorecards that can help people to define and measure success: Key metrics can be matched to specific strategies and then shared, tracked, and discussed. Users can create meaningful visualizations that convey the right information the first time, aggregating content from multiple

sources and displaying it in a web browser in an understandable and collaborate environment. Moreover, rich interactivity allows users to analyze up-to-the-minute information and work with data quickly and easily to identify key opportunities and trends (Figure 14).

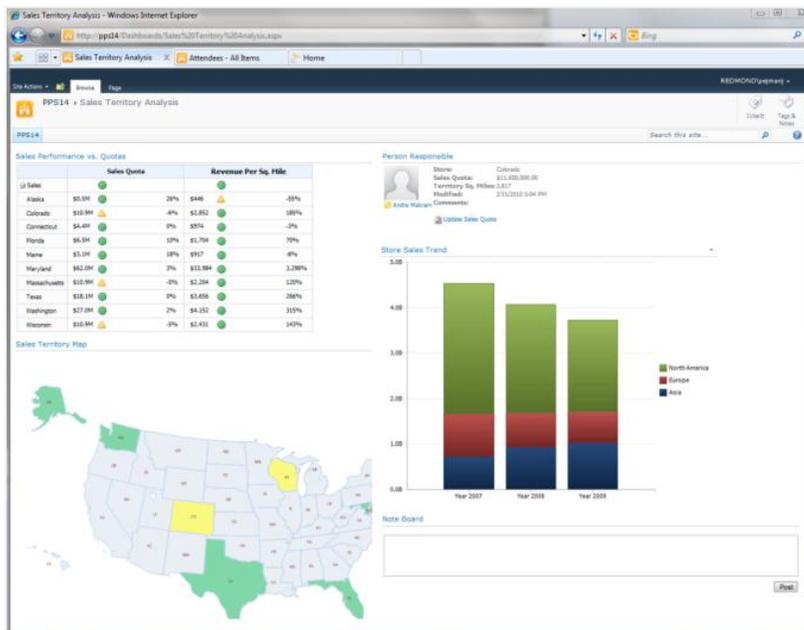


Figure 14: Dashboard in SharePoint Server 2010 Insights

Conclusion

Many organizations today face the problem of having critical business data locked in siloed legacy systems. Your organization can resolve this problem by connecting its LOB systems to SharePoint Server 2010, which provides both business and technological benefits. Workers from across the organization can use a single interface to access appropriate data more quickly and easily, helping to increase the ROI of legacy systems and decrease the need for training. In addition, the native application development capabilities of SharePoint Server can help to speed the time-to-market of your solutions, ultimately freeing IT resources to focus on more strategic initiatives.

SharePoint Server 2010 provides many options for building solutions that integrate business-critical data from disparate systems. Two of the most powerful tools in SharePoint Server are Business Connectivity Services and Enterprise Search. These tools can provide unprecedented access to data from external sources, and they can display this data in a highly interactive and visual manner—all on a unified platform that is easier to manage and more secure than legacy applications.

To design a seamless SharePoint Server-to-LOB connection, IT architects should consider the following four key steps. It is a good idea to plan all of these steps before beginning:

1. Identify the business process that needs improvement.
2. Establish which LOB application (or applications) you are targeting.
3. Determine what data is appropriate to bring into SharePoint Server.
4. Decide how you want users to interact with the data once it is in SharePoint Server.

Planning around these four steps can help IT architects to define the most appropriate connectivity method based on business goals. The fictional example of Fabrikam Manufacturing in this white paper demonstrates each of the four steps in action, with an emphasis on the technical capabilities of Business Connectivity Services.

Importantly, a core benefit of connecting to external business-critical data is flexibility: Once data resides in SharePoint Server 2010, it can be viewed and manipulated in numerous ways. For example, additional business processes can be implemented or extended using out-of-the-box platform capabilities; productivity can be improved using advanced capabilities for visualizing data in real time; and success can be defined and measured using interactive dashboards and scorecards.

Resources

Learn more about the SharePoint Server capabilities outlined in this white paper by visiting the following:

- [Overview of Business Connectivity Services](#)
- [Business Connectivity Services Technical Resource Center](#)
- [Creating Custom Business Connectivity Services Connectors Using SharePoint Server 2010](#)
- [Composite applications](#) and [composite application development](#)
- [SharePoint Server Partner Finder page](#) (ISVs and SIs by area and vertical who can help to architect and deploy SharePoint Server solutions)
- [DIRA framework](#) (role-based productivity with SharePoint Server in discrete manufacturing)