



**APPLIEDINFORMATIONSCIENCES**

**TITLE:**

**SharePoint Development Platform – How the SharePoint Server 2010 enhancements further enrich its application development platform credentials**

**May 12, 2010**

## TABLE OF CONTENTS

About AIS .....	3
Introduction .....	4
Maturation of SharePoint 2010 as an Application Development Platform .....	5
Application Layers.....	9
Presentation Layers .....	10
Business Services Layer .....	13
Data Access Layer .....	15
Other Noteworthy Platform Improvements .....	18
Records Management .....	19
Service Applications.....	19
Microsoft Office Documents .....	20
Business Intelligence .....	21
Access Services .....	22
Development Tools Improvements .....	22
Development Tools .....	23
Development Environment .....	25
Solution Deployment.....	26
Conclusion.....	26
References .....	27
About the Authors .....	28



## ABOUT AIS

Since 1982, Applied Information Sciences, Inc. (AIS) has provided software and systems engineering services to government agencies and enterprise companies looking to modernize existing applications, automate business processes, gather business intelligence, and create more effective workflow solutions for their employees, partners and customers.

For over 15 years AIS has been a Microsoft Managed Gold Partner. AIS has Microsoft certified-competencies in custom .NET development, Microsoft Office SharePoint Server, and Microsoft Business Intelligence solutions. With experience dating to inception, AIS is considered to be one of the premier Microsoft application development partners with .NET and SharePoint platform core competencies. Our established record of envisioning, designing, building, and delivering solutions that exceed client expectations – on time and within budget – is why Microsoft chooses to recommend AIS as a trusted solution provider again and again.

AIS is very active in the Microsoft technology community, regularly contributing to thought leadership discussion and participating in invitation-only Microsoft Partner Advisory Councils. AIS is also part of the Microsoft Regional Director program.

## INTRODUCTION

In early 2007, AIS wrote the white paper [Microsoft Office SharePoint Server 2007 \(MOSS\) as an Application Development Platform](#) {1}, making the case that SharePoint was an excellent platform for building .NET applications. Since then AIS has successfully built large scale, mission critical applications on the SharePoint platform for a number of clients {2}[LNK1] including the Federal Bureau of Investigation, the Department of Defense, Towers Watson, CITGO Petroleum Corp, and Fannie Mae. We even built and certified the DOD 5015.2 Add-On Pack for Records Management for MOSS – and we consider it to be as good of an example of an application built on top of SharePoint as any.

In his 2008 keynote address at the Microsoft SharePoint Conference, Bill Gates indicated that the number of licenses for Microsoft Office SharePoint Server 2007 (MOSS) had surpassed 100 million {3}. Through the variety of extensibilities available – including Workflow, Custom Lists, Event Handlers, InfoPath Forms, Web Services, and Integrated SQL Server Reporting Services – more and more IT managers are leveraging SharePoint to build custom applications. With the momentum of broad adoption behind it, the business world has embraced SharePoint as more than just a portal product but as a platform for building collaborative web applications.

Microsoft's release of Microsoft SharePoint Server 2010 marks the continued maturation of SharePoint as a development platform for building even richer collaborative web applications and integrating with other Line of Business systems in the enterprise. The purpose of this white paper is to review the primary reasons to consider leveraging SharePoint as an application platform and to get a glimpse into SharePoint 2010's new features and enhancements that better support this assertion. Along the way, we will compare and contrast the application methods in SharePoint 2010 with those in SharePoint 2007 (MOSS). This paper is based, in part, on our work as part of Microsoft's Technology Adoption Program (TAP) for SharePoint 2010.

Please note that core concepts described in this white paper are either SharePoint Server features or are features that are inherited from underlying technologies on which SharePoint Server is based, including SharePoint Foundation 2010 (the free version of SharePoint formerly known as Windows SharePoint Services) and ASP.NET 2.0/3.0/3.5. We will use the term SharePoint to collectively refer to these features.

The primary audience for this paper includes architects, IT managers, and consultants involved in building rich, collaborative web applications. The secondary audience is technical decision makers who want to make the business case for portals and collaborative web application investments. This paper assumes a basic understanding of high-level SharePoint concepts. For the readers who may not be familiar with the SharePoint concepts, we encourage you to review the earlier version {1} of this paper, before proceeding.

## MATURATION OF SHAREPOINT 2010 AS AN APPLICATION DEVELOPMENT PLATFORM

Let us begin by briefly reviewing an example we included in the previous paper. Figure 1 depicts a small application built on SharePoint Server 2007 for managing a list of webcasts.

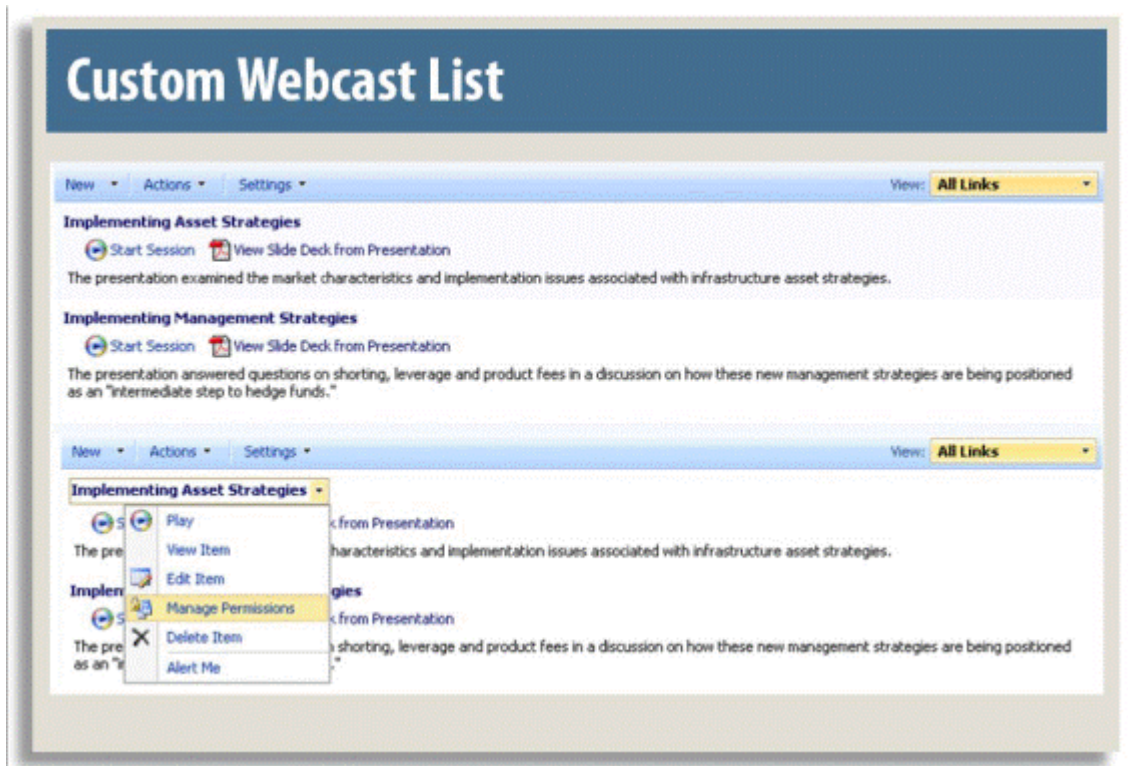


Figure 1:

While it may not seem difficult to add a list control on an ASP.NET page and hook up some ADO.NET code to persist the data in the database, it should be noted that we did not write a single line of code. We relied solely on SharePoint list handling and the content database to implement the list. But SharePoint 2007 goes beyond this. Imagine that we are required to extend beyond the simple functionality above. For example, a requirement is that each item in the list be individually secured. Additionally, users may also want to subscribe to any changes to the list (new recordings) made via RSS or email. From a QA standpoint, a content management process needs to be enforced when a new webcast recording is uploaded that requires versioning, check-in/check-out and approval workflow. As such, a content management requirement invariably necessitates the ability to maintain an audit trail of changes as well as the ability to undelete an item that is inadvertently deleted. Last but not least, a search function on the site should include the information about the webcasts. With the need for these additional features, a custom ASP.NET solution becomes more difficult. Fortunately, all of these functions are provided by the SharePoint List by default. We can even extend the SharePoint List behavior using event handlers and custom fields. All of the information stored inside a list is accessible, not only via the SharePoint UI but also via the Object Model (Class Library as well as Web Service based OM). This means that processes outside the host process can access the list information, which is key to building transparent applications that are also reusable.



The webcast list example above exemplifies the following three key benefits of building on SharePoint:

- 1) Ability to leverage the platform building blocks (such as SharePoint lists) to speed up the development;
- 2) Empowering the end-users by allowing them to directly customize the list and add content; and
- 3) Taking advantage of the out-of-the-box governance features of SharePoint such as an audit trail, enforcing quota limits and the backup/restore.


Over the last three years, AIS has successfully built a number of applications by taking advantage of the aforementioned out-of-the-box features and also by extending the platform using custom tools and utilities on top of the SharePoint 2007. In doing so, we not only developed deep insights into the platform capabilities but also ran into some challenges. The main challenge is related to the integration of external data sources. Any data that needs to be surfaced within SharePoint 2007 has to reside within the content database. Since application developers are not allowed to directly access the data residing in the content database, they are limited in adding functionality, such as reporting that typically requires direct access. The other key challenge is caused by the gap between the developer and the end-user tools. For instance, it is not possible for an end-user to create a workflow in SharePoint Designer and have it be imported within VS.NET for more advanced customization. Finally, the challenge related to the tooling and the development environment. All the development for SharePoint 2007 has to take place on a server Operating System such as Windows Server 2003, and even though SharePoint 2007 is squarely based on ASP.NET, the tooling is not on par with vanilla ASP.NET-based development.

Fortunately, SharePoint 2010 seeks to alleviate each of the aforementioned challenges. SharePoint 2010 supports the latest .NET 3.5 platform enhancements (such as WCF and LINQ) in addition to Silverlight 3.0. The Business Data Catalog (BDC) has become Business Connectivity Services (BCS) and has moved into the core SharePoint Foundation platform. BCS supports full read/write capabilities, client side caching, and is the basis for Entity Backed Lists, the mechanism by which you can create SharePoint Lists on top of a database table (or other external data sources). On top of the platform building blocks SharePoint provides, the development tools receive major improvements that enable developers to take advantage of these new features (as well as features that were found in MOSS 2007) with much less effort. SharePoint Designer 2010 receives a major upgrade, supporting the creation of more complex workflows that are more re-usable. SharePoint Designer will become *the* design tool for building Business Connectivity Services entity definitions. Visual Studio 2010 comes with SharePoint 2010 tools that far exceed the extensions available for Visual Studio 2005/2008. Web Parts and other SharePoint pages can be built with graphical editors and debugged without leaving the Visual Studio environment. Feature and Solution packaging receive much improved tools for preparing the meta-data required to deploy to SharePoint. The experience is much more like that of ASP.NET web applications.

Based on the improvements described above, we believe that the following use cases should be considered when building applications on the SharePoint 2010 platform:

1. The need to provision more than one website based on a logical grouping – such as department, region or country – rather than have one website that serves all users.

For example, a company needs to develop a web application for its partners that allow them access to pertinent sales information. It would be perfectly reasonable to start out with an ASP.NET application; however, as the application usage grows, partners would like to customize the site based on their own SharePoint Server unique needs. The partners may want the sales information to surface differently (i.e. grouped by regions vs. grouped by cities), or they may want to co-locate additional applications on the same page (i.e. a tax calculator). Rather than building all of this personalization in code, it is easier to provision a site for each partner that is based on a



single common site blueprint. Each partner can then customize their site based on their specific needs via a single code base.

2. The need to manage un-structured content (i.e. documents, web casts, etc.) combined with social networking capabilities.

Most modern websites need to manage ever increasing digital content. A distinction between the structured content and unstructured content is that the former deals with data that can be viewed and managed using set-based groupings (database views), whereas the latter deals with data that is managed as explicit standalone entities along with the associated metadata. Not only is it important to manage the integrity and security of standalone entities (such as documents); it is also important to manage the relationships among them (for instance, the association relationship between a document and structured application data). Additionally, provisions need to be made to facilitate business process workflow (routing, approval, regulatory compliance, etc.) and document and folder-level access control and search. The recent trend of social networking makes unstructured content even more challenging. Users are demanding the ability to tag a document, rate it, locate and interact with the subject matter experts related to the document, all in near real-time. Again, rather than building these capabilities into an application, it is far easier to rely on the Content Management Services provided by SharePoint.

3. Organizations seeking to involve the business users and analysts in all aspects of development, not just during the requirements gathering process.

One of reasons for the success of SharePoint has been its ability to empower end users. From creating custom views on lists, customizing web parts and building workflows, SharePoint has allowed the endusers to create much of the functionality they need without relying on IT each step of the way. However, in many instances this has led to organic growth that can get out of control easily. By enhancing the collaboration between these roles and providing a model for better IT governance, SharePoint 2010 makes further inroads in this area. Consider the following example: it is possible for the business user to customize a live site inside the browser and then save the site as a template. A developer can then import this template into a development tool such as Visual Studio and continue to build on the customization made by the business user. The end product is then available again to the business users.

## SharePoint Tools Continuum

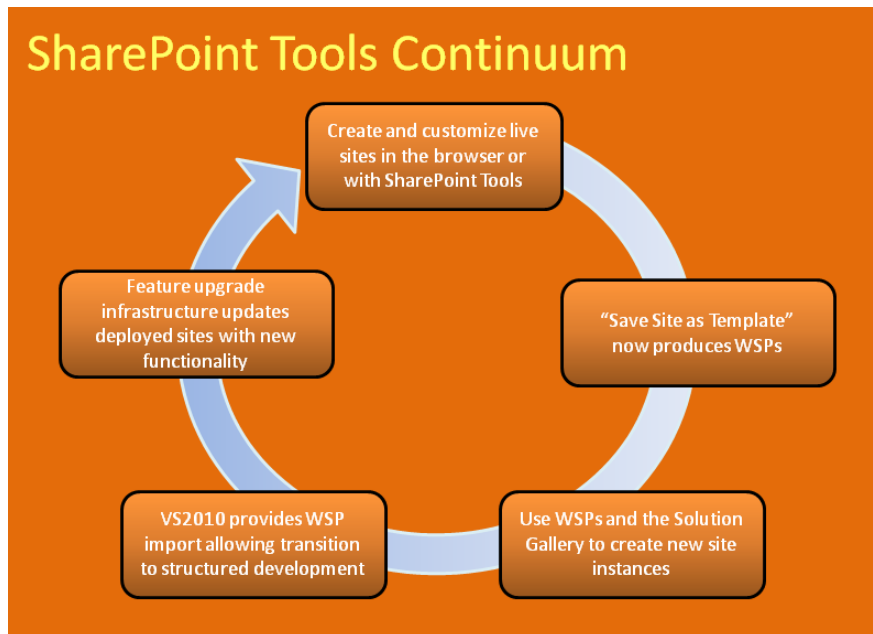


Figure 1: SharePoint 2010 Tools Continuum

#### 4. Organizations looking to build departmental applications that are based on external sources of data.

While there are a number of approaches for integrating LOB applications within SharePoint Server 2007, a number of challenges remain. With the inclusion of BCS in SharePoint 2010, it is now possible to surface external data sources as SharePoint lists, generate CRUD screens and incorporate external data into the integrated search. External data sources are not limited to relational data but can also include Web Services and plain .Net classes. By incorporating various departmental applications into a single platform, organizations can reduce the total cost of ownership through one common user experience, development and deployment strategy. Another major innovation is the addition of Access Services, designed to allow organizations to move their existing Access based application to SharePoint Server 2010.

In an attempt to show how these use cases can be effectively realized, we will review the various SharePoint Server 2010 enhancements in detail in the remaining sections of this paper.

## APPLICATION LAYERS

Like any development platform, SharePoint 2010 provides platform elements for building a layered application. The following diagram (Figure ) illustrates how these technologies stack up.

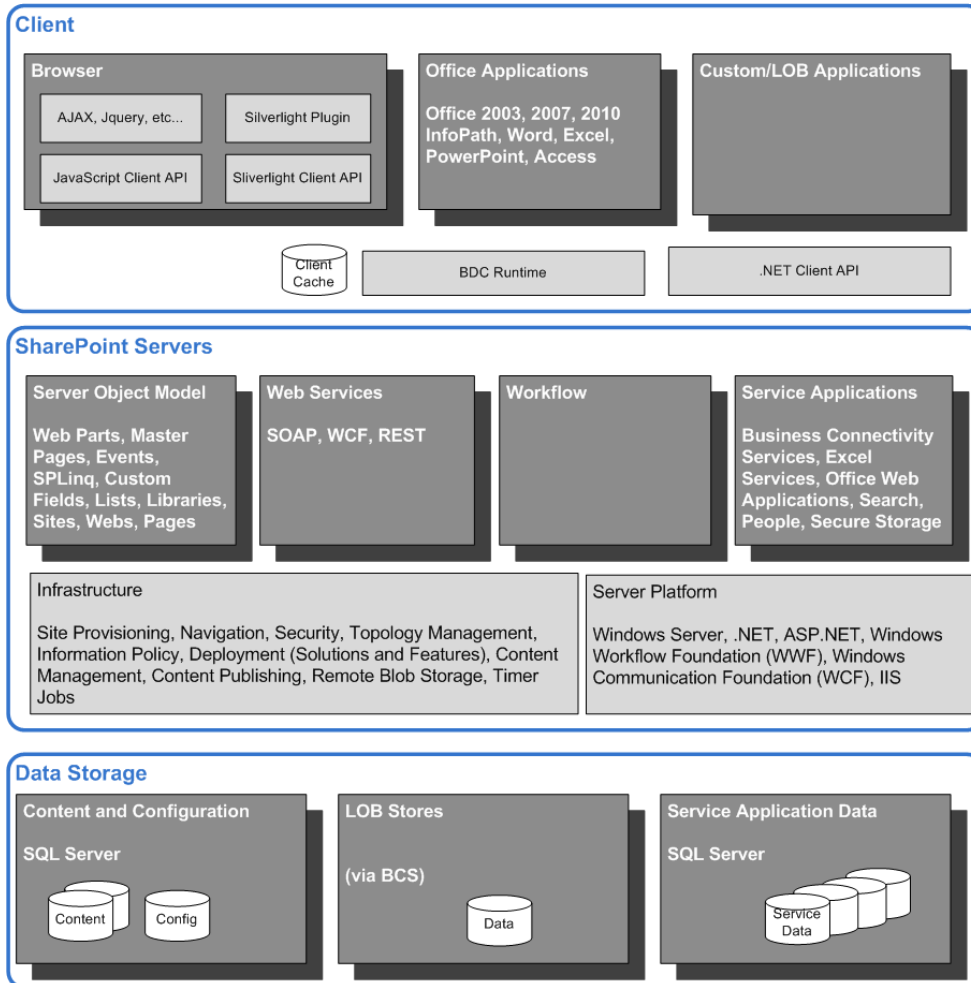


Figure 3: SharePoint Technology Stack

At the bottom of the stack is the data storage layer, where SharePoint stores its content and configuration data, where SharePoint application state is persisted, and where Line of Business (LOB) systems store data. For the SharePoint portion, this is SQL Server; for other systems, it could be a variety of other data stores with different access mechanisms (SQL, Web Services, XML, other business systems). Above this is the SharePoint server, a 64-bit Windows Server layer with the .NET Framework, ASP.NET and IIS. At this layer SharePoint implements core functionality for the SharePoint application platform, which includes sites, navigation, security, pages, web parts, document libraries, lists, etc. In addition to web site hosting, service applications operate on this layer, providing additional functionality such as search, Business Connectivity Services and people (User Profiles). The server object model, web services and service applications provide the mechanism and interfaces that allow application development to occur. At the top of the stack lies the primary means by which users interact with SharePoint – the client layer. Most commonly this will occur through a web browser viewing SharePoint pages.

SharePoint 2010 provides a richer browser-based experience for users than SharePoint 2007 by utilizing JavaScript and Silverlight. Microsoft Office Applications have deep integration with SharePoint, allowing interaction with SharePoint without leaving the business applications that business users work within each day. The same client side capabilities utilized in the browser and in Microsoft Office are available to other applications that can make use of SharePoint. Client side caching and synchronization is fully integrated with the SharePoint Server, providing the functionality to Microsoft Office for the sometimes disconnected user.

## PRESENTATION LAYER

A major labor saving component of SharePoint 2010 is the Presentation Layer. Built into SharePoint on top of ASP.NET, are the building blocks of the User Interface: navigation, Master Pages, Page Layouts, web parts, custom field controls, List Property Edit and Display Forms, and List Views. All of these features are there for the developer to build on, customize and extend with varying degrees of effort. Over the three years since the release of SharePoint 2007, some customizations that could be made were often too difficult to implement to be practical solutions. Some menu features were hidden in hard to find places; list edit and display forms were hard to customize; CAML (Collaborative Application Markup Language) notation for list views were not a skill familiar to many developers; and many menu and navigational actions caused the browser page context to change. Microsoft has gone to great lengths to address some of these areas in addition to introducing User Experience innovations gleaned from research and experience with the Office products (both 2007 and 2010) and Windows Presentation Foundation.

### Ribbon

One of the most obvious new features of SharePoint 2010 is the introduction of the now familiar Office 2007 style ribbon to most pages. This ribbon stays at the top of the page and does not scroll out of view. Common tasks are presented in a manner that makes it easier to find what you may want to do with fewer clicks.

The ribbon can be customized by adding custom actions or removing SharePoint options.

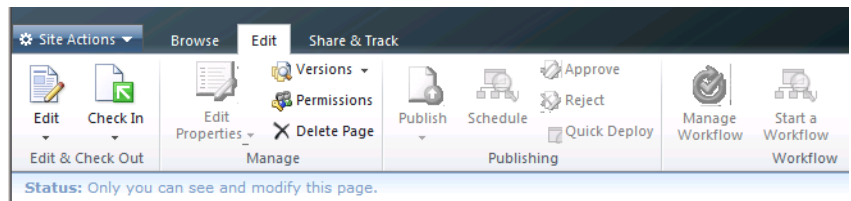


Figure 4: Ribbon and Status Bar in SharePoint 2010

### Status Bar and Notification Area

Just below the ribbon is an area called the Status Bar that can display status messages and action links (see Figure 4 above). Below the Status Bar is a Notification Area (see Figure 5) that provides a mechanism to display transient mentions for a period of time (default of 3 seconds).

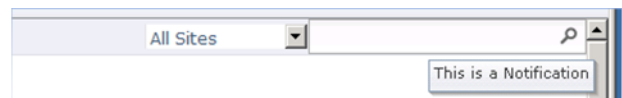


Figure 5: SharePoint 2010 Notification Area

## Dialogs

Another fairly obvious new feature of SharePoint 2010 that is visible when you look at an item's properties is the new pop-up Dialog capability. Rather than navigate away from a page, these dialogs are displayed in an AJAX style popup window allowing the users to remain in the context of the primary page they were viewing without navigating the browser away from that page (see Figure 6).

Developers can use the Ribbon, Status Bar, Notification Area and Dialogs to inject functionality that fits seamlessly and acts the same way that SharePoint itself does.

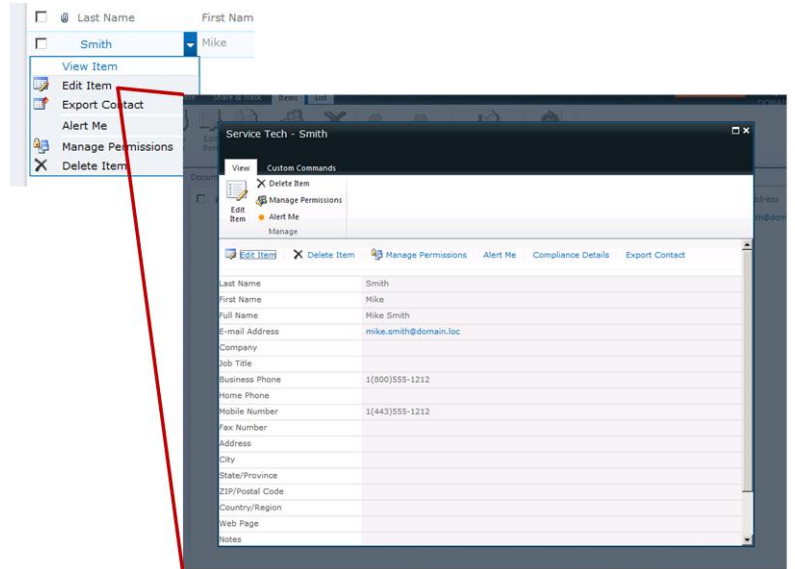


Figure 6: View Item Dialog Box in SharePoint 2010

## Customize Forms with InfoPath

A major labor saver in SharePoint for Lists and Libraries is the automatic generation of UI for Create, Read, Update and Delete operations. SharePoint automatically generates forms for displaying and editing item details. With SharePoint 2010, this is even true for External Lists, making it possible to generate these forms for database tables through BCS.

In SharePoint 2007, customizing a List Item's form for editing could be done in several ways, such as SharePoint Designer or custom List Definitions and Visual Studio. With SharePoint 2010, forms can be customized with another very useful form designer, InfoPath 2010. With InfoPath 2010 forms, the developer has a number of options for designing the form with simple form layout, multiple views, complex validation logic and access to additional data sources to augment the editing process.

## XSLT Views

A powerful feature of SharePoint has been the ability to create different views of a List in much the same way you might create views on a database table. In SharePoint 2007, a developer could create new views using CAML. In defining a view for a list, CAML provides semantics for translating list data into HTML. For many developers, CAML is a foreign language, and it is not a skill shared by a large group of people. A much more widely known and used method for transforming data into HTML is XSLT. With SharePoint 2010, Microsoft has introduced XSLT-based Views as the preferred mechanism for defining list views. These views can be applied to standard SharePoint 2010 Lists and Libraries as well as to External Lists. The flexibility of XSLT coupled with a large base of developers already possessing XSLT skills makes creating custom views a much more economical option.

## Silverlight

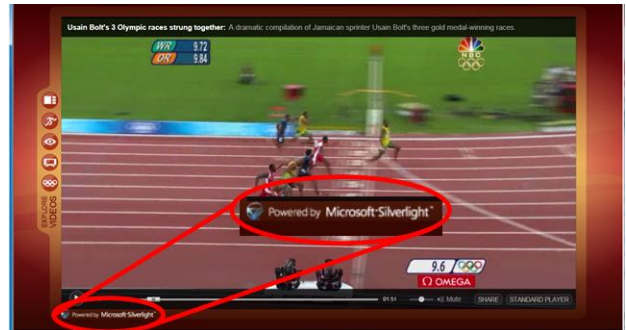


Figure 7: Silverlight Video Player for 2008 Beijing Olympics

Silverlight is a lightweight version of the .NET framework designed for the Internet {4}. Silverlight contains a wide variety of ready to use rich controls such as TreeViews, Charts, Grids and Auto-Complete text boxes, with a growing number of third party controls available for free or for pay. Silverlight also provides support for the latest streaming media formats and has become a common mechanism for video on the web. Silverlight was used successfully for the 2008 Summer Olympics in Beijing by NBC {5} (see Figure 7).

This technology allows developers to build applications for the Web in Visual Studio and Expression Blend using much the same programming skills for building Windows applications. Contrast this with the significant differences between building ASP.NET applications and building Windows applications. By providing the rich user experience offered by Silverlight on the web, many difficult programming tasks in ASP.NET and JavaScript become much simpler and allow the same set of skills to be applied to Windows and Web applications.

Microsoft provided a blueprint for building Silverlight components and integrating them with SharePoint 2007 in early 2008. {6} SharePoint 2010 however, combines with Silverlight to provide a very powerful platform for building Rich Internet Applications (RIA), with SharePoint providing the host web site, security and data storage. SharePoint 2010 ships with a Silverlight Web Part that allows any Silverlight application to be dropped quickly into a SharePoint site. Much of what was demonstrated in the Silverlight Blueprint for SharePoint 2007 is delivered with SharePoint 2010 or made even easier. SharePoint 2010 itself ships with Media Viewer and Organization Hierarchy components built with Silverlight.

Common functionality that can be implemented with SharePoint include snappy navigation, simplified viewing and editing of complex data, providing streaming audio or video and stylish visual presentation and effects. While much of this was also possible with SharePoint 2007, SharePoint 2010 provides

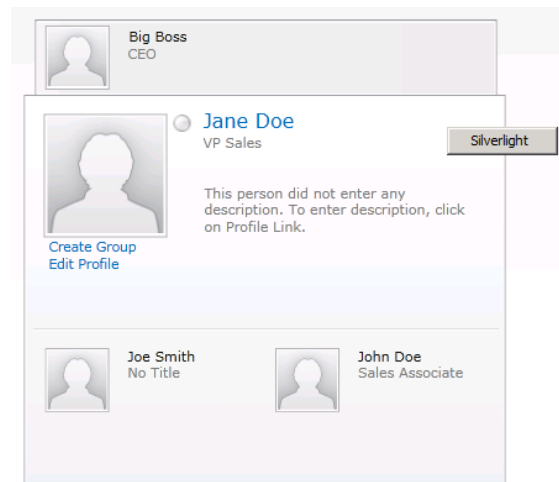


Figure 8: Organization Browser

significant new power through a better client side API. This API provides a programming model on the client (in the Silverlight application) that is almost identical to the one SharePoint developers have become familiar with on the server. The objects, properties and methods are almost the same. The new API encapsulates the data access between the client and the server, making it very easy to build SharePoint-aware Silverlight components that can access SharePoint 2010 Site and List data.

## Client Object Model

For developers who may need to write code that does not execute on the SharePoint Server, SOAP web services are the primary remote interface in SharePoint 2007. In addition to a much improved Web Service capability, SharePoint 2010 provides a Client API that wraps those web services and provides an Object Model that more closely mirrors the Server Side Object Model. While this object model is a subset of the Server

```
string siteUrl = "http://MyServer/sites/thesite";
ClientContext clientContext = new ClientContext(siteUrl);
SP.List oList;
oList =
clientContext.Web.Lists.GetByTitle("Announcements");

ListItemCreationInformation itemCreateInfo;
itemCreateInfo = new ListItemCreationInformation();
ListItem oListItem = oList.AddItem(itemCreateInfo);
oListItem["Title"] = "My New Item!";
oListItem["Body"] = "Hello World!";
oListItem.Update();

clientContext.ExecuteQuery();
```

Figure 9: Sample Announcement Creation Code

Object Model, it contains the most commonly used objects (Webs, Lists, List Items, Content Types, Fields) necessary to perform a wide variety of programming tasks. The Client API provides a unified object model for the Silverlight CLR, .NET CLR and JavaScript, providing access in almost any client scenario. This capability allows for developers with one set of skills to apply those skills to both the Client and the Server.

## BUSINESS SERVICES LAYER


SharePoint 2007 includes a number of middle-tier capabilities of interest to developers. Developers can access SharePoint via a Server Side Object Model as well as a variety of remote options, such as SOAP based web services, FrontPage Remote Procedure Call (RPC), Web Part Page Services, URL Protocol (owssrv.dll) and WebDAV. Workflow and Event Handlers allow business logic and data manipulation to occur at specific points in the life cycle of a SharePoint content item. The variety of remote options and the disparity between client and server programming has made it difficult to learn all of the possible ways to perform an action. SharePoint 2007 workflows created with SharePoint Designer 2007 are hard to reuse outside of the List they are created for and they have to be tied to a List (or Content Type). Some desirable points in the life cycle of a Web, List or Item aren't supported by the Event framework. SharePoint 2010 has significant improvement in all of these areas.

## Rest APIs

A major new improvement to how developers access SharePoint List and Library data is through REST (Representational State Transfer {7}) APIs. REST defines a simplified protocol for querying and retrieving data from web services that has become popular in the last few years. Based on WCF and ADO.NET Data Services (also called Astoria), these new services provide a much simpler programming model than the SOAP based List Data Web Service in SharePoint 2007. Using Visual Studio, it is possible to use data binding against these services in Silverlight, Windows Forms applications, and WPF applications.

## Workflow

Workflow is one of the most powerful features of SharePoint 2007. In addition to the out-of-the box workflows that are included with SharePoint 2007, it is possible to use the SharePoint designer to build declarative (no-code) workflows. For more advanced scenarios, developers can create custom workflows in Visual Studio. Two of the main challenges in building workflows with SharePoint 2007 relate to the limitations in SharePoint Designer 2007 and the workflow host that is part of SharePoint 2007. While SharePoint Designer allows business



users to develop simple workflows, it requires a direct connection to the site for which the workflow is being developed, thus limiting the ability to reuse the workflow in different locations. Furthermore, there is no easy way to leverage code components developed by IT. The workflow host within SharePoint 2007 comes pre-configured with a set of services such as persistence, eventing and tracing. There is no way to add custom services to the runtime host. For instance, there is no direct way to allow SharePoint-based workflows to interact with events other than the list item events.

SharePoint 2010 alleviates these challenges, allowing the workflows created using SharePoint Designer to be saved as templates for reuse. SharePoint Designer 2010 has also been enhanced significantly to allow business users to leverage tools such as Visio for workflow modeling and visualization. Another major improvement is the ability to modify out-of-the-box SharePoint workflows (e.g. approval workflow and three-state workflow). Additionally, the workflow host within SharePoint 2010 now provides the extensibility option to inject custom services. It is now possible to kickoff workflows without explicitly associating it to a list item. Finally, from an overall scalability perspective, it is possible to designate application server nodes as workflow participants. In other words, unlike SharePoint 2007 where each application server node participates in executing the workflows, in SharePoint 2010 there is a way to throttle the workflow execution to a limited set of machines. This allows isolation of services within the farm so system administrators can better manage resources and troubleshoot issues.

### Improved Events

SharePoint 2010 has also greatly enhanced the Event capabilities. A major improvement is the introduction of “After-Synchronous” events that execute before the user is returned control after saving an item. A common issue with event handlers in SharePoint 2007 is the fact that the event handler is not finished before the user sees the updated item in a List View, which can present race conditions, or at least confusing behavior for the user. If the event handler modifies the data in the item, users may not see this right away or until they have refreshed the page. In SharePoint 2010, After-Synchronous events also allows for providing custom pages for displaying errors. Imagine an event handler that wants to tell the user that the newly updated item has an impact or a conflict with another item or that an attempt to write to a database failed. Without a custom page (as in SharePoint 2007), the event handler would throw an exception with a single text message that would be displayed on a plain, generic error page. The custom pages in SharePoint 2010 could show the user details about the other item(s) in conflict or provide the user links or actions to perform to address an issue raised by the event handler.

Site scoped events are event handlers that are registered for an entire Site rather than requiring individual registration on each List in each Web in the Site Collection. With SharePoint 2010, new events for Web Creation and List Creation are available for managing changes to sites and performing actions on those objects after they are created. For example, a specific set of groups might need to be added to a Web when it is created or a List of a certain type might need to be registered with a global synchronization mechanism that monitors a set of lists across many Webs or Site Collections. It could be as simple as notifying administrators or site owners of new things being created, updating a reporting database or recording additional audit information.

### DATA ACCESS LAYER

One of inherent capabilities of SharePoint is the collection of data. SharePoint-based applications capture data via user input through multiple means, such as metadata on list and document items, InfoPath forms and workflow interaction. Many enhancements have been made in SharePoint 2010 to improve the validation and

integrity of data as it is collected. New capabilities have been added so that developers can better access and interact with the data as well as building applications that respond to changes that occur with the data.

## Lookup Columns

A commonly used tool in SharePoint 2007 has been Lookup Columns. These columns provide the ability to present a dynamic set of values for users to choose when entering data into an Item. With SharePoint 2010, the Lookup Column has been improved to allow multiple columns to be selected to go along with the primary column SharePoint 2007 provided (see Figure 10). These additional columns are available to List Views as well as displayed in the Detail view of an item's properties.

Lookup Columns defines either a 1-many or many-many relationship between two lists in much the same way as two tables in a relational database. SharePoint 2010 has introduced the ability to define a Cascading Delete or Restrict Delete constraint on Lookup Column relationships (see Figure 11). A list manager can indicate that when a value in a list that is referenced by a Lookup Column is deleted, any items in the list the Lookup Column is in are deleted. The list manager can also indicate that delete not allowed if it is referenced by items in the other list.

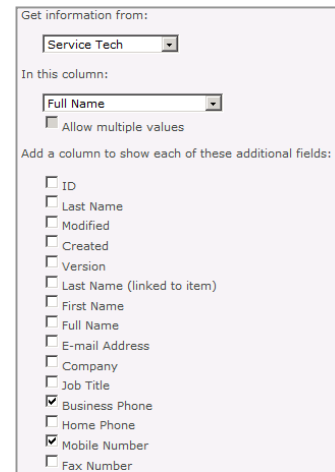


Figure 10: Lookup Column Configuration

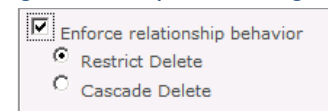


Figure 11 - Defining Relationship

## Validation and Unique Values

Another improvement to standard lists is the ability to add validation formulas to each field in a list. The formula for a field can be created to return True if the validation criteria is met or to return False if not. A configurable text message will be displayed in the Edit form if the validation formula returns False.

For more complex validation that needs to access multiple columns (for example to compare Start Date to Due Date), a list manager may also add an Item level validation formula (see Figure 12). This validation formula is set in the List Settings and can access the value of any Field in the item.

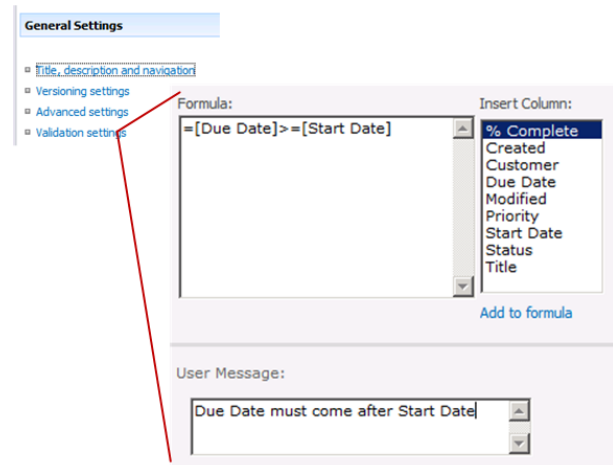


Figure 12: Item Validation

SharePoint 2010 also provides the ability to restrict duplicate values in a column by setting "Allow Duplicate Values" to "No".

All validations are enforced by the SharePoint platform, not in the Presentation Layer. Required fields, unique fields and fields with validation formulas are enforced for all types of access.

## Large List Support

Microsoft has put much emphasis on performance of large lists in SharePoint 2010. This is realized in several new features: Query Throttling, Batch Query Support, Remote Blob Storage and External Lists. Query Throttling prevents queries against large lists from consuming too many resources and taking too long. By default, SharePoint now blocks queries that return more than 5000 rows (a configurable amount), unless it is using an Indexed column that supports the query – or if it asks for too many joins.

In order to support developers

accessing large numbers of items in lists, SharePoint 2010 includes a new

API for batch processing. The *ContentIterator* class shown in Figure 13 returns a configurable number of items from a query in a loop, allowing developers to concentrate on what they want to do with the data rather than how to get a large number of items without having a query timeout or taking up too much server memory.

In SharePoint 2010, Remote Blob Storage provides a mechanism to store document bits outside of the SharePoint Content databases. With SharePoint 2007, a key factor in sizing and managing farms is managing Site Collection (and therefore Content Database) size. Document Storage, one of the most useful and widely used capabilities of SharePoint, is the primary reason databases grow beyond manageable sizes. To address this issue, Remote Blob Storage in SharePoint 2010 provides the ability to place these files outside of the SQL Server database, such as on file shares (SANS, NAS, RAID arrays, File Servers). Microsoft will provide a Remote Blob Storage provider with SharePoint 2010 and SQL Server 2008 R2, but other vendors may also provide capabilities tailored to different types of storage or that integrate with other types of Content Management Systems.

External Lists are another mechanism by which large lists can be incorporated into SharePoint. These lists are part of Business Connectivity Services described elsewhere in this paper. External lists can be managed as separate data in separate data stores, such as SQL Server, outside of the Content Database. By being outside of SharePoint, External Lists can be optimized and tuned individually. This allows skilled developers and DBAs to provide very large lists without growing SharePoint 2010 content databases.

## LINQ to SharePoint

LINQ is a technology introduced in .NET 3.5 {8} that provides Object Oriented access to SQL like queries directly in the .NET language (C#, VB.Net). LINQ to SharePoint extends this capability to SharePoint List Data, unifying the data access

```
ContentIterator ci = new ContentIterator();
string query = @"<View><ViewFields>
<FieldRef Name='Title' /><FieldRef
Name='Completed' /><FieldRef Name='CompletedBy' />
</ViewFields> <Query>" +
ContentIterator.ItemEnumerationOrderByID +
@"</Query></View>";

ci.ProcessListItems(
    list,
    query,
    1000, // Batch size of 1000
    true, // Recursive query
    delegate(SPListItemCollection items) {
        // Do something with batch of items
    },
    delegate(SPListItemCollection items, Exception e
    {
        // Handle exception
    }
    )
    )
```

Figure 2: Iterating Through A Large List Sample

```
from o in data.Orders
where o.Customer.City.Name == "London"
select new { o.Amount, o.Customer.Name };
```

Figure 3: Sample LINQ Query

model that can be used for all types of data. With this new capability comes the ability to retrieve data from multiple Lists through joins on Lookup Column values. Figure 14 illustrates C# code to retrieve data from three lists: Orders (Amount), Customers (Name), City (Name). Retrieving this type of data using the SharePoint 2007 object model would be a lot more lines of code, and is a completely different programming task. A .NET programmer familiar with LINQ for SQL Server, but not an experienced SharePoint developer, would be able to use LINQ for SharePoint quickly.

## Business Connectivity Services

In SharePoint 2007, the Business Data Catalog, or BDC, makes it possible to display business data in web parts, add Business Data Columns to lists, import Business Data into User Profiles and find Business Data with SharePoint 2007 Search. All of this capability is bound to the Enterprise version of SharePoint 2007, which costs more and leaves some organizations without this very useful tool. With SharePoint 2010, the BDC's successor, Business Connectivity Services (BCS), some of these capabilities are included in SharePoint Foundation and, therefore, available to all SharePoint 2010 flavors (see Figure 15). This alone is a major improvement over

SharePoint 2007, but there are also significant new capabilities found in BCS. One limitation of the SharePoint 2007 BDC is that it is read/only. It provides the ability to add "actions" that can navigate the user to the source application or a custom application to perform edits to the data. In SharePoint 2010, BCS supports full CRUD operations on this data (if desired, you can still use it in a read/only fashion). BCS also now supports Windows Communication Foundation (WCF) services and .NET classes in addition to the database and SOAP based web services supported by BDC.

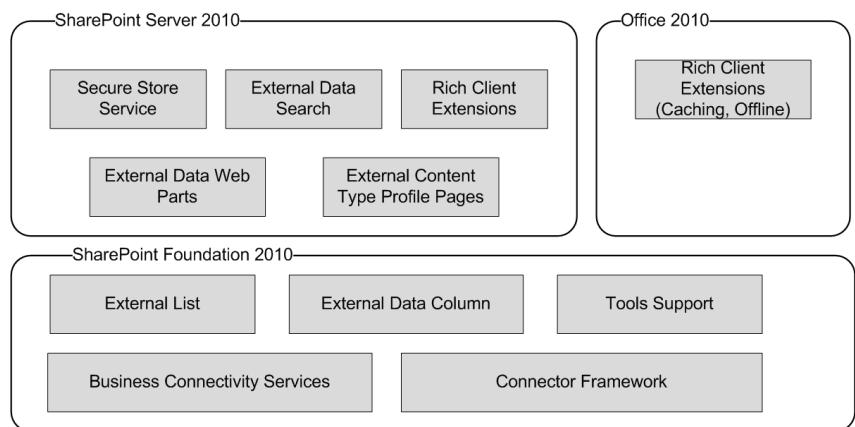


Figure 4: Business Connectivity Services Features

In the SharePoint 2010 BCS, the External Content Type (ECT) encapsulates the business entity and metadata describing that entity. A BCS metadata model defines Entities in much the same way BDC does in SharePoint 2007. Each of these Entities represents an External Content Type that can then be used by SharePoint 2010 or Office 2010. As announced at the 2008 Microsoft SharePoint conference {9}, SharePoint 2010 now supports creating entire lists directly on top of database tables. This capability is realized by creating an External List from an External Content Type. An external list contains auto-generated forms and views for displaying, editing, creating and deleting items. Not only does this provide a more complete way to surface business data in SharePoint than custom web parts or Data View web parts, but it also exposes the data to Enterprise Search and provides some control over the scale and performance of large lists.

BCS can access data directly via a database instance, by invoking a WCF or SOAP-based web service-based interface, or by executing methods on .NET classes. In BCS terminology, each external application is a Line of Business System (represented in metadata by <LobSystem>). Discrete objects such as Customer that are part of external applications are defined as Entities. The operations that can be invoked on Entities are represented as Methods. Methods can be of different types including a Finder method (that return instances of an Entity) and Specific Finder (that return a specific instance of an Entity). The relationship between Entities, such as master detail, can be modeled as Associations.

Central to the SharePoint 2010 BCS architecture is the notion of a metadata that defines how the external data source can be integrated into SharePoint. Metadata contains information on how the external data can be accessed. BCS can access the data directly via a database instance or by invoking a web service-based interface provided by the external system. In BCS terminology, each external application is a System. Discrete objects (roughly nouns) such as customers that are part of external applications are defined as Entities. The operations that can be invoked on Entities are represented as Methods (see Figure 16). Methods can be of different types including a Finder method (that return instances of an Entity) and Specific Finder (that return a specific instance of an Entity). The relationship between Entities, such as master detail, can be modeled as Associations. Figure 16 – BCS Metadata Example illustrates an example of syntax for BCS metadata, which is very similar to the BDC metadata syntax found in SharePoint 2007 but with some changes and additions.

```

<Entity Name="Customer">
  <Methods>
    <Method Name="GetCustomer">
      <MethodInstance Type="Finder" .../>
    </Method>
  <FilterDescriptors>
    <FilterDescriptor Name="Region" Type="Comparison" />
  </FilterDescriptors>
  <Parameters>
    <Parameter Name="Region" Direction="In">
      <TypeDescriptor .. AssociatedFilter="Region"/> ...
    </Parameter>
  </Parameters>
</Methods>
</Entity>
<Association Name="CustomerToOrder"
  AssociationMethodEntityName="Customer"
  AssociationMethodName="GetOrdersByCustomer">
  <SourceEntity Name="Customer"/>
  <TargetEntity Name="Order"/>
</Association>

```

Figure 16: BCS Metadata Example

Figure 16 – BCS Metadata Example illustrates an example of syntax for BCS metadata, which is very similar to the BDC metadata syntax found in SharePoint 2007 but with some changes and additions.

With SharePoint 2007, this metadata XML is either hand-crafted or built with a few tools that eventually became available from Microsoft or third parties. However, with SharePoint 2010 Microsoft provides full BCS designer capability in SharePoint Designer 2010 as well as support in Visual Studio 2010 for building .NET data sources. These tools provide full packaging and deployment support as SharePoint Solutions. Simple applications that point to basic database tables can be created in minutes with SharePoint designer.

Authentication and authorization is key to enabling any type of access to LOB systems. The SharePoint 2010 BCS provides a number of ways to ensure that business data is secured. All data connections, whether database, SOAP, WCF or .NET classes can be authenticated to the LOB system in a number of ways, using either the current user, a process identity (such as a SharePoint Web Site Application Pool) or stored credentials (stored securely in the Secure Store Service, the new version of SharePoint Single Sign-on). The BCS metadata model also allows securing access to External Content Types at multiple levels, from an entire LOB system to an Entity to a specific method of an Entity.

Another key feature of BCS is the client side runtime. This consists of a client-side component that maintains a local store of BCS data and metadata. The client-side runtime maintains this local store as a cache and manages two-way synchronization with the data source. Office 2010 client applications make use of this store to access the same External Content Types used in SharePoint 2010. While this runtime is delivered to the client via the Office 2010 product, the Client API would be available to custom applications that wish to make use of BCS in a fashion similar to the way Office 2010 uses it.

## OTHER NOTEWORTHY PLATFORM IMPROVEMENTS

## RECORDS MANAGEMENT

SharePoint 2010 drastically improves upon the capabilities for managing records by extending where and how records are managed. Multiple records repositories are now fully supported within a single SharePoint farm, as is the ability for users to send content to multiple repositories. When sending content, rules can be set up to define whether or not the file is copied to the repository, moved to the repository or moved with a hyperlink left in the original location. Records managers can define routing rules to aid in classifying records that use content metadata to determine where a record should be placed within the archive. Records repositories now support the use of a hierarchical file plan and the ability to apply retention rules based on location within the file plan.

In addition to using a repository, one of the new capabilities of SharePoint 2010 is the ability to declare records in-place, applying the appropriate retention rules, policies and security to the content as it sits within a library. This allows for content to be appropriately managed from its current location without the need to move it to a repository. Users have the ability to declare records from directly within the library via the toolbar (see Figure 17).



Figure 17: Record Declaration Button

SharePoint 2010 provides flexibility for defining and applying retention rules, whether records are managed in-place or within a repository, by allowing for multi-stage and recurring retention schedules. Recurring retention schedules allow for an action to recur periodically, such as periodic review of vital corporate records. Multi-stage retention schedules allows for more advanced management of records by providing the ability to define rules such as “Maintain 3 years after project completion, transfer to archive for 5 years, then destroy”.

These capabilities are part of the SharePoint Server platform, providing organizations with flexibility and capabilities to have “compliance everywhere” within their SharePoint implementation.

## SERVICE APPLICATIONS

In SharePoint 2007, the Shared Services Provider (SSP) concept allows a collection of common services, such as search, user profiles, my sites, Excel Services, etc, to be provided to one or more Web Applications. This allows for the sharing of information, capabilities and hardware without unnecessary duplication. However, the SSP model could result in an unnatural coupling of services and web sites. A SharePoint 2007 Web Application has to subscribe to all of the services in the SSP and can use only one SSP. Inter-Farm capabilities exist in SharePoint 2007, but they are not friendly to WAN type connections for large or global organizations.

SharePoint 2010 frees the farm from being restricted to specific sets of Shared Services by providing a model where the individual services, now called Service Applications, can not only be subscribed to by Web Applications as necessary, but where many can be connected across a WAN. This allows for Geo-Distributed farms such as one where a single, global People Service (User Profiles) can be consumed from farms on different continents (see Figure 18). In SharePoint 2007 this requires an aftermarket capability to replicate this User Profile information to multiple SSP’s and smart re-directs to send users to the correct farm for their My Site.

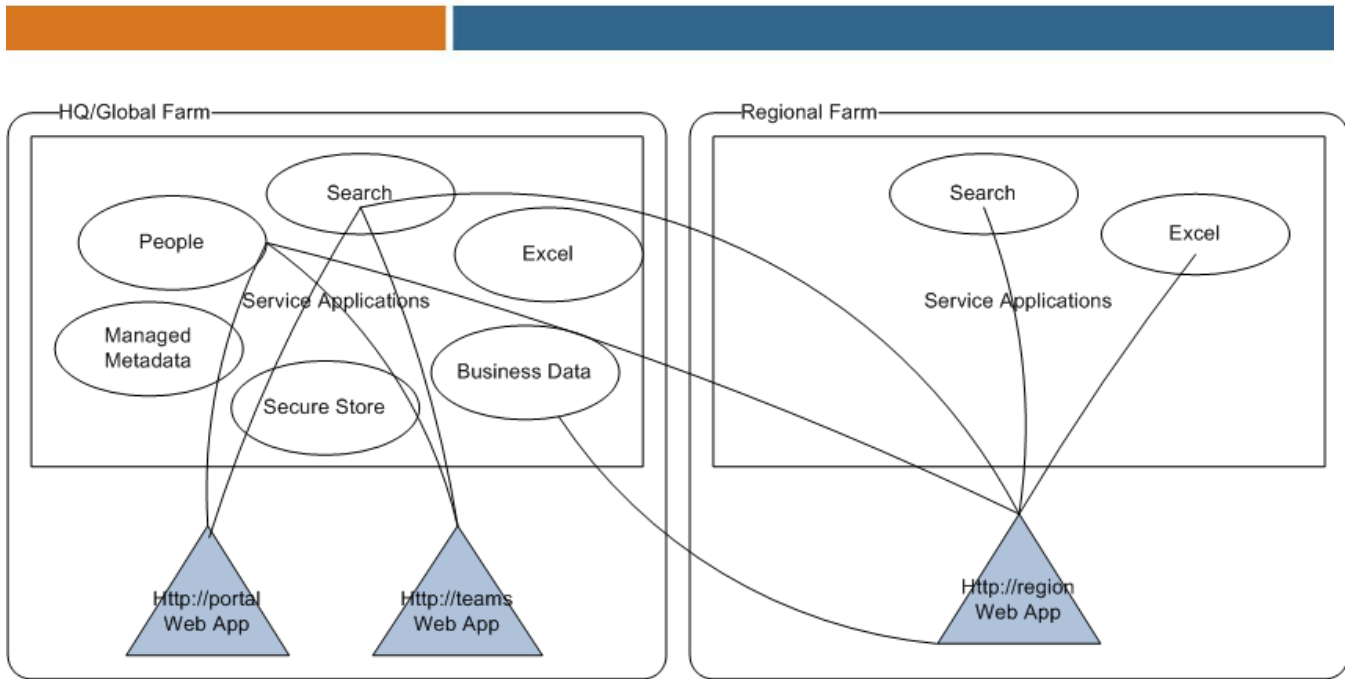


Figure 18: BCS Metadata Example


Custom Service applications can be built to provide functionality that takes advantage of the Service Application architecture for deployment, administration and load balancing. Imagine a business application that requires web services, timer jobs, a persistent data store, an administrative interface and can make use of other SharePoint Service Applications, such as the Secure Store service. By hosting this kind of application as a SharePoint Service Application, the platform and functionality of SharePoint provide the foundation and the developer can focus on the business value of the application without spending a lot of time on the basic building blocks already available from SharePoint 2010.

### MICROSOFT OFFICE DOCUMENTS

Starting with Office 2007, Microsoft has transitioned the file format used by the Office applications (Word, Excel, PowerPoint) to the Open XML format, which, in a nutshell, is a compressed zip file consisting of a folder structure containing mostly XML files. For prior versions of Office, generating or manipulating content within these files required using the API exposed by the Office applications themselves. Doing this on the server has proven to be difficult to achieve and manage, especially in large scale. With the advent of the Open XML format, however, server-side file manipulation became much more feasible.

From a development perspective, the Open XML Format SDK {10} is available to search, manipulate, extract and generate documents in this format easily. Combining this with the native capabilities of SharePoint 2010, such as document management, workflow and event notification, many document automation scenarios can be addressed with ease. These solutions range from automatically generating a PowerPoint presentation based on information contained with SharePoint lists to assembling a Word document from multiple document parts and sending the final document through an approval process.

SharePoint 2010 has the capability to host Office Web Applications for editing Word, Excel and PowerPoint files within the browser. Office documents generated or manipulated as described above can then be viewed or edited directly on the Web without the need for client-side software. Whether or not users have Word or Excel, they can consume the information in these files. They don't even need to download the entire file to their computer (though they can do that if they choose). Charts or extensive calculations can be performed in Excel with user input combined with business data from an LOB system. PowerPoint slide decks can be generated from



manufacturing or sales data. Proposals or Quotes can be auto generated, modified by multiple users, and declared a record during a workflow without ever creating additional copies in e-mail or on someone's hard drive. While some of this can be achieved with the Open XML SDK in MOSS, SharePoint 2010 makes it easier and offers a much more complete user experience.

## **BUSINESS INTELLIGENCE**

Business Intelligence is another area with significant improvements in SharePoint 2010. To begin, there are a number of scalability improvements to the desktop version of Excel 2010 including support for large number of rows that goes far beyond the 64K limit in the current version. The new version of Excel is 64-bit and can take advantage of multi-core computers to greatly speed up the calculations.

Microsoft has made a strategic decision to leverage Excel as the primary BI analyst tool. As part of the self-service BI initiative, Excel users - through the Power Pivot capability - can import large amounts of data from various data sources and quickly generate pivot tables. Once the data is imported into Excel 2010, using a feature called "Slicers", users will be able to filter the data easily. In other words, Excel users can extract, transform and load (ETL) data from multiple sources directly into Excel without requiring IT to build a formal data import setup. The workbook containing the imported data can later be shared with other users by publishing it to SharePoint 2010. Under the covers, the publishing process creates an SQL Server Analysis Services instance that can be monitored by IT, using tools such as usage dashboard and resource quota.

Excel Services, which is the server-side Excel functionality provided by SharePoint Server, has seen a number of improvement as well. First, it directly benefits from the scalability improvements to the underlying calculation engine mentioned earlier. Second, two new object models have been added to the existing Excel Services API: a JavaScript-based client-side object and the new RESTful API. The JavaScript-based client-side object can be used to trap events and drive the UI – for example it will be possible to trap an event raised when a user edits a certain cell inside the workbook and then scrolls the workbook to a certain range. The RESTful API can be used to embed charts and ranges contained inside a workbook, on a web page outside of SharePoint.

Finally, scorecard, dashboard and analytical capabilities from Office Performance Point Server 2007 will continue to be available within SharePoint 2010.

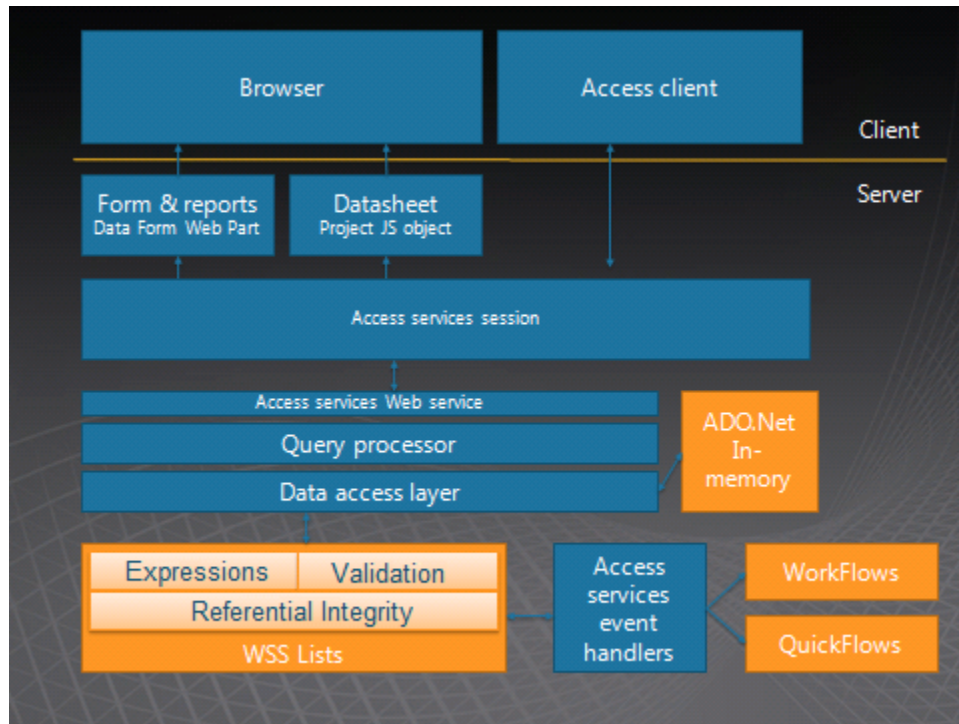
## **ACCESS SERVICES**

Microsoft Access has been a very popular development tool in the Microsoft Office family for more than a decade. It combines the power of a relational database engine with easy to build data entry forms and reports. In earlier versions of SharePoint, Access provided the ability to connect to SharePoint list data (as a linked table), which provided some integration between the tools. However, users couldn't host the data file in SharePoint for a central location for all of the data, forms, reports and business logic to be shared among many users.

With the advent of Access Services in SharePoint Server 2010 and Microsoft Office Access 2010, there is now a true browser-based version of that valuable Microsoft Access application. Access 2010 has the ability to publish an Access database and application to a SharePoint 2010 site running Access Services. This converts the tables to lists, creates the appropriate web forms and even publishes reports as SQL Server Reporting Services reports.

The user experience is a web site that looks and feels like an Access application, but in the browser. At the same time, the Microsoft Access client can be used to connect to and manipulate the data and update the application.

The figure below depicts the architecture of the Access Services Service Application. The browser and Access client interact with the Service Application. Browser-based Access utilizes auto-generated forms and reports. Table design and validation logic is automatically implemented in SharePoint lists. Event handlers and workflows provide business logic implementation.



Access Services provides an extremely easy to use mechanism to implement custom applications with data with a tool that has been a favorite for a large number of information workers and developers for a long time. While it doesn't offer the complete set of capabilities of the Access 2010 client by itself, Access Services will prove to be an extremely valuable tool for application development on the SharePoint platform.

## DEVELOPMENT TOOLS IMPROVEMENTS

As an application platform, Microsoft provides multiple tools to configure and customize SharePoint 2010. These tools provide an array of capabilities for a variety of user roles, based on the complexity of the application being created. The tools include a browser, InfoPath 2010, SharePoint Designer 2010 and Visual Studio 2010. Combined, they provide the ability to create powerful enterprise applications that leverage the capabilities of the SharePoint platform quickly.

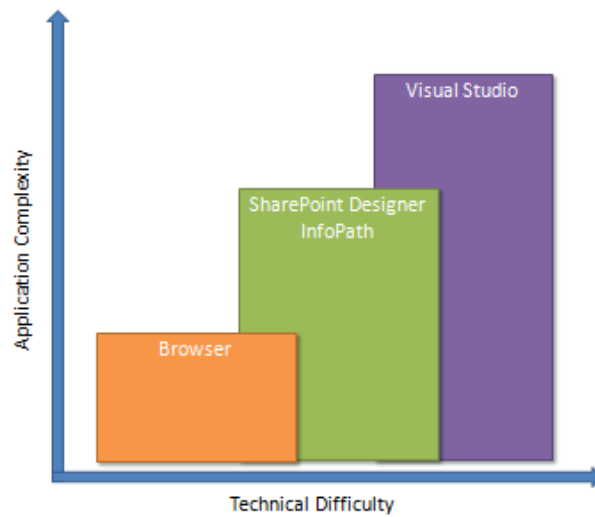


Figure 5: SharePoint Development Toolset

The platform also provides a consistent format for packaging the customized solution functionality, whether it be a single web part or a highly customized site template, which is leveraged by the development toolset. Also, multiple deployment options exist depending upon the scope of the solution and the level of administrative involvement required.

## DEVELOPMENT TOOLS

### Web Browser

SharePoint 2010 provides users with the capability to use a web browser to configure SharePoint entities such as lists, workflows, and webs, in such a way that they can be reused and re-deployed within other SharePoint locations. Users can configure a SharePoint site to contain specific lists, content types, workflows and predefined data through the browser and then save the site as a template (see Figure 20). The site template can then be made available for other users, and when a site is created from this template, it will look and contain the same SharePoint entities as the original site. Using the browser to save templates allows an easy means to create consistent and re-deployable SharePoint entities with no coding involved.

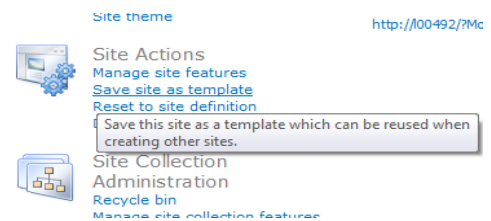


Figure 20: Site Template Creation

### InfoPath

InfoPath Designer 2010 can be used to create applications quickly that allow users to interact with SharePoint 2010, external list data and SharePoint workflows. Users can design forms to replace the existing SharePoint list, external list and workflow forms with greater control over the business rules, data validation and formatting of the information. InfoPath also allows for multiple views of the same data and the ability to hide or display parts of the form based on business rules. To get started, InfoPath Designer includes predefined form templates so users designing forms can generate new forms quickly and easily. Also included are predefined lookup fields, which allow the ability to lookup information from another SharePoint list. Customized forms can also include the pre-population of form data and the ability to interact with data from multiple lists.

Once published to SharePoint, the forms generated in InfoPath Designer can be displayed within the InfoPath client application or displayed in a web browser via Forms Services. InfoPath forms can leverage the existing SharePoint web part infrastructure, allowing forms to not only be hosted within web parts, but also to communicate and share data with other web parts on the page. Additionally, forms published to SharePoint 2010 are portable. Saving a list as a template that uses a custom InfoPath form as a solution file ensures that the InfoPath form will be used when new lists are created from the template.

## SharePoint Designer

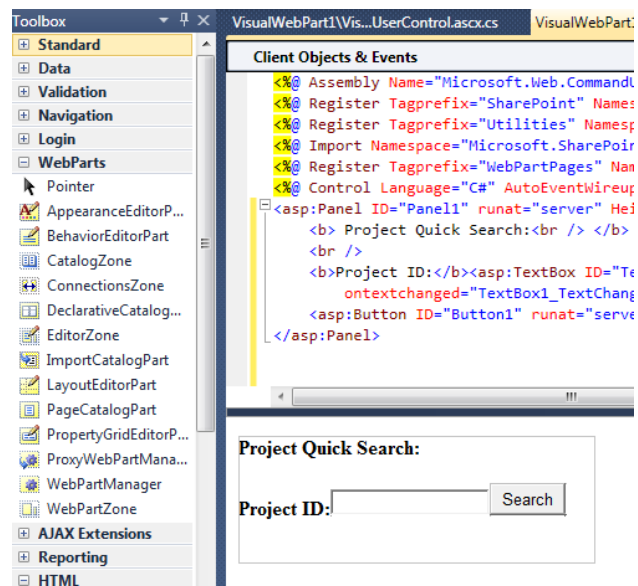
SharePoint Designer 2010 has been much improved. New capabilities for the creation of BCS entities, list schemas and custom actions have been added; major improvements have been made with regard to designing workflows and editing pages, and new workflow actions are made available. SharePoint Designer 2010 leverages the WSP solution package format, allowing SharePoint Designer 2010 to take advantage of configurations made by a browser and additionally, allows artifacts generated by SharePoint Designer 2010 to be imported in to Visual Studio 2010. SharePoint administrators also have greater control over how SharePoint Designer 2010 is used within their environment. Administrators can block the ability to modify master pages and also prevent SharePoint Designer 2010 from being used at either the site collection or web application level.


## Visual Studio 2010

Visual Studio 2010 contains extended capabilities for developers to create rich applications on the SharePoint 2010 platform. Many improvements have also been made in Visual Studio 2010 to increase developer productivity as well as take advantage of SharePoint 2010 functionality. These improvements not only help make it easier for .NET developers to create SharePoint solutions, but also to package, deploy and integrate SharePoint artifacts created by multiple tools into one central environment.

Improvements have been made to Visual Studio 2010 so that developers can set up projects and write and debug code more easily. Project templates are available in Visual Studio 2010 for common SharePoint project development items such as web parts, event receivers, workflows and list definitions. These templates set up the project so that developers can start writing code more quickly, adding stubbed-out class files, assembly references and making the proper configurations, such as those required for a Sandboxed solution. The templates also aid in the deployment and debugging processes by automatically creating and deploying the feature and solution packages, resetting the application pool and attaching to the correct process for debugging when the developer hits f5.

New visual designers have been added to aid in the development of workflows, building BCS models, creating application pages and designing features and solution packages. These designers reduce complexity and increase productivity for both SharePoint and .NET developers. Also, connections to SharePoint 2010 sites can be made directly from the Server Explorer, allowing developers a common view into all of the components that make up the SharePoint site.





Although SharePoint development is much more than just the creation of custom web parts, web part development has long been an area where the experience between SharePoint development and ASP.NET development somewhat diverged. However, with Visual Studio 2010, the development experience for the creation of web parts has been vastly improved. Developers now have the ability to visually design, create, deploy and debug SharePoint 2010 web parts in the same way as ASP.NET User Controls (see Figure 21).

Artifacts created from other applications can be imported into Visual Studio to be extended and/or placed under source control. For example, declarative artifacts created in SharePoint Designer 2010 can be

Figure21: Visual Web Part Designer

imported to Visual Studio 2010 and extended with code and added to a Team Foundation Server (TFS) source code repository. Once integrated, the capabilities of TFS can be leveraged to provide full Application Lifecycle Management for SharePoint 2010 team projects, including team and continuous builds, traceability of requirements through code check-ins, adherence to and compliance with coding best practices and visibility into the overall health and stability of the code base.

To further accommodate the requirements of specific projects, the templates, connection nodes and project deployment steps are all customizable and extensible.

## DEVELOPMENT ENVIRONMENT

There are now several options for setting up the environment necessary for developing solutions for SharePoint 2010. As with previous versions of SharePoint, a local environment is necessary when using Visual Studio 2010. However, for development purposes, SharePoint 2010 can now be installed on a client operating system (Vista SP2 or Windows 7) in a supported manner, assuming the client operating system is 64 bit. As with previous versions, SharePoint 2010 can run either directly on hardware or within a virtualized environment.

## SOLUTION PACKAGES

SharePoint 2010 uses a single solution format for packaging and deploying customized functionality. The solution package is a compressed file containing all of the components necessary for the customized solution, such as features, web parts, list definitions, assemblies, customized ASPX pages, workflows and all other components. The solution package provides a consistent means for packaging and deploying customized solutions, independent of the functionality contained within the solution. Using a single, defined format allows for easy transport of artifacts between solution development tools. For example, a list can be created and customized in SharePoint Designer 2010, then imported into Visual Studio 2010 and further extended.

Many solutions contain customized lists, libraries and content types. In the previous version of SharePoint, these customizations can only be done through extensive modification of XML files, with the laborious task of also creating the solution package for deployment. In SharePoint 2010, these customizations can be done very quickly through point and click configuration with SharePoint Designer 2010. Once done, the customizations can be imported into Visual Studio 2010, which will automatically create a deployable solution package with these customizations. .

A visual designer for the creation and maintenance of feature and solution packages is included in Visual Studio 2010. This provides developers the ability to visually define, control and maintain the solution packages, such as artifacts that make up the features – features within a solution package, feature deployment scope, safe control definitions and feature dependencies.

## SOLUTION DEPLOYMENT

In the previous version of SharePoint, solution packages are deployed via a farm administrator using the stsadm command-line tool. This method of deployment is still a viable option in SharePoint 2010. However, SharePoint 2010 provides an additional deployment method - referred to as Sandboxed Solutions - for solutions that have a defined scope and can be safely deployed without having to involve the farm administrator.

### SandBoxed Solutions

Sandboxed solutions inherently provide security and manageability to customized solutions by isolating the scope to a site collection and only allowing custom code access to a subset of the SharePoint object model. Developers still have the ability to create common SharePoint components needed for many solutions, such as web parts, event receivers and workflows, but are not able to access resources outside of the context of the site collection. Additionally, farm administrators can define quotas to ensure sandbox solutions do not consume resources in excess of specified thresholds. This helps to ensure that custom code does not inadvertently bring down a SharePoint 2010 farm by consuming too many resources.

Deployment of a SandBoxed solution is done by uploading the solution package to a Solution Gallery within the site collection, similar to uploading a document to a library. This can only be done by a user that is an administrator of the site collection. Once uploaded, the site collection administrator can activate the solution and make the functionality available within the site collection. Any features that are deployed by the solution can then be activated within the sites of the site collection by the site administrator(s).

## CONCLUSION

SharePoint's out-of-the-box capabilities coupled with its ability to behave as an application development platform have proven to be a hit and the adoption of SharePoint continues to grow in organizations of all sizes. SharePoint 2010 will continue this trend by providing improved out-of-the-box features, greater scalability and entirely new and easier ways to integrate custom, enterprise scale applications with intranets, portals, extranets, internet sites and personal sites. SharePoint 2010 provides the core platform services and you, the developer, provide the domain-specific customizations and functionality critical to your organization's success.

## REFERENCES

- [1] "Microsoft Office SharePoint Server 2007 (MOSS) as an Application Development Platform"  
<http://www.appliedis.com/ideas.html>
- [2] Applied Information Sciences Customers  
<http://www.appliedis.com/work.html>
- [3] "Bill Gates Touts Strong Growth and Enterprise Adoption of Microsoft Office SharePoint Server 2007"  
<http://www.microsoft.com/presspass/press/2008/mar08/03-02SPPTEnterpriseGrowthPR.mspix>
- [4] Microsoft's Silverlight Home Page  
<http://www.silverlight.net/>
- [5] Silverlight at Beijing Olympics  
<http://www.microsoft.com/Presspass/Features/2008/jul08/07-29qaolympics.mspix>
- [6] Silverlight Blueprint for SharePoint  
<http://www.codeplex.com/SL4SP>
- [7] Representational State Transfer (REST), Roy Thomas Fielding  
[http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)
- [8] LINQ MSDN Site  
<http://msdn.microsoft.com/en-us/library/bb308961.aspx>
- [9] SharePoint Conference 2008 Keynote Transcript  
<http://www.microsoft.com/Presspass/exec/billg/speeches/2008/03-03SharePoint2008.mspix>
- [10] Open XML Format SDK 2.0  
[http://msdn.microsoft.com/en-us/library/bb448854\(office.14\).aspx](http://msdn.microsoft.com/en-us/library/bb448854(office.14).aspx)



## ABOUT THE AUTHORS

Alex Holcombe  
Senior SharePoint Architect  
Applied Information Sciences

Jim Mullennix  
Senior SharePoint Architect  
Applied Information Sciences

Vishwas Lele  
Chief Technology Officer  
Applied Information Sciences