

Chapter

2

Introducing Information Bridge Framework

Overview

Information Bridge Framework is one of Microsoft's answers to business intelligence problems. Information Bridge is a technology that provides the gateway between LOB services and Office documents. Information Bridge Framework is the set of tools, applications, services, and application programming interfaces (APIs) that allow users to access LOB services from Office documents, including Word documents, Excel spreadsheets, InfoPath forms, and Outlook e-mail messages, as well as from Web pages opened in Internet Explorer.

Architecture

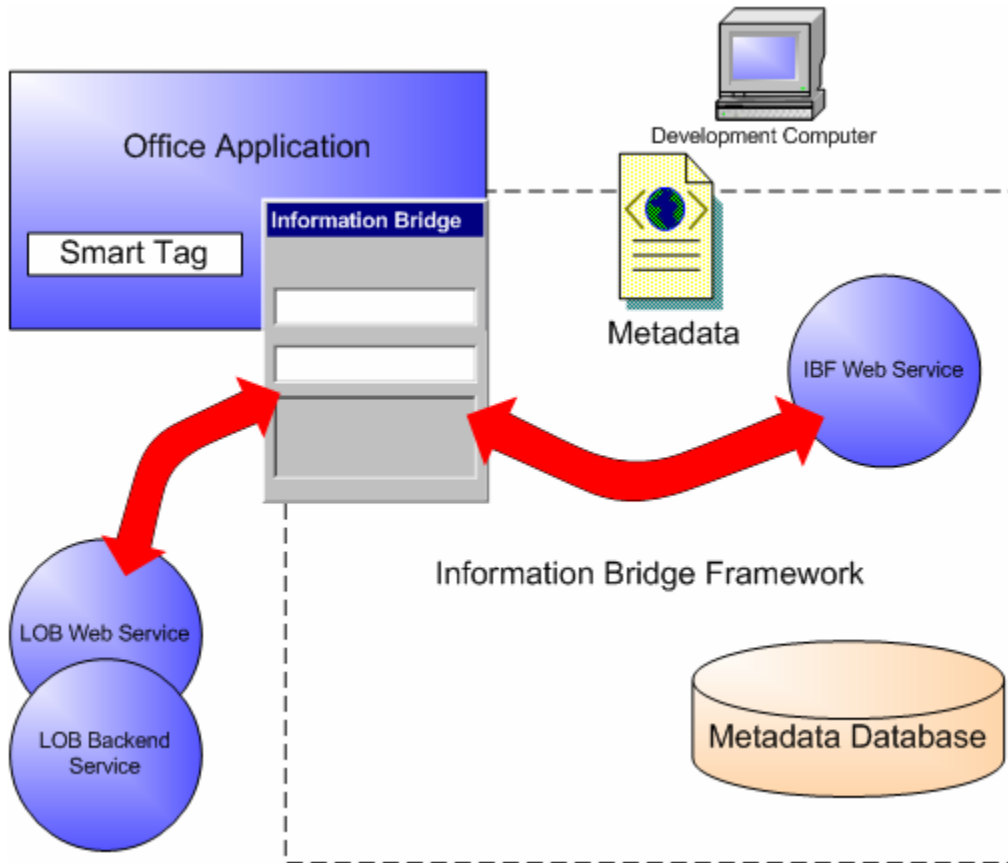
The Information Bridge architecture consists of a client component that connects to backend LOB services through either Web services or common language runtime (CLR) reflection, with the former mechanism being the most common. The client component is represented as the Information Bridge information window, which is usually hosted in the Document Actions pane in Word documents. The following figure shows the Information Bridge information window:



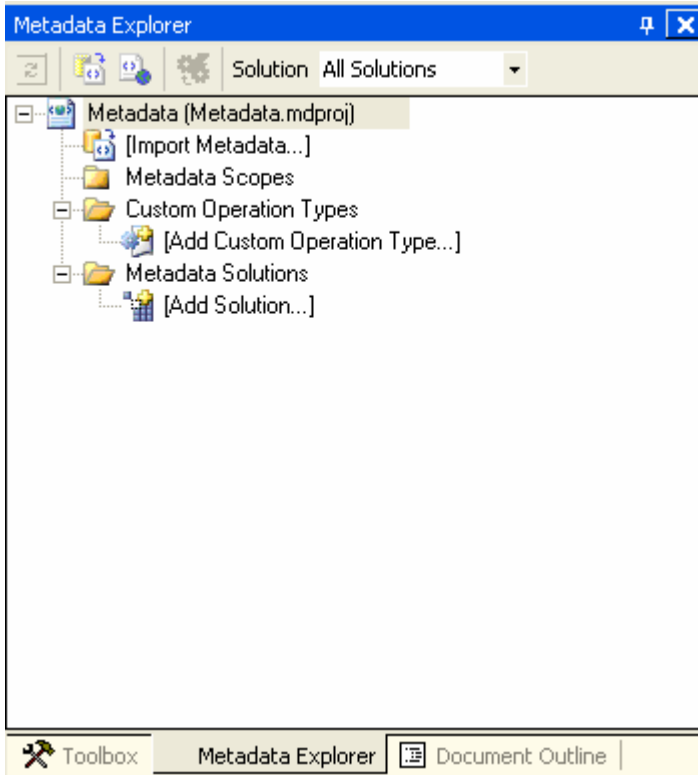
The client component accesses LOB services by means of a metadata service. The Information Bridge metadata service is an XML Web service that returns information that describes how the Information Bridge client component should connect to the backend services and display returned data to the end user. The Web service uses Authorization Manager for security.

Information Bridge Framework also consists of Information Bridge Metadata Designer, which is a Visual Studio .NET plug-in that helps developers create metadata, publish it, and retrieve it from the metadata service. Once metadata has been published, it can be reused in multiple Information Bridge solutions.

Here is a pictorial view of the basic architecture:



The developer uses Metadata Designer (not shown) to create metadata that describes the actions that can be taken from within the Information Bridge client component. Each action encapsulates one or more operations to retrieve, manipulate, or display LOB data. An action depicts an end-user mechanism for retrieving, manipulating, and viewing that data. For example, an executive could execute the “GetClientInfo” Information Bridge action, which could in turn invoke multiple operations on multiple data sources. However, to the executive, only a single action would have been performed. The developer can accomplish this using the new Metadata Explorer tool that is available with the new Metadata project type in Visual Studio .NET. Here is a picture of Metadata Explorer:



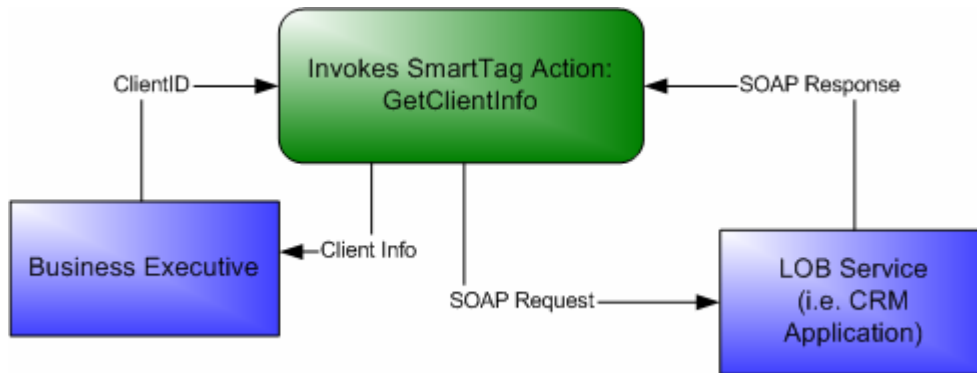
The developer is also responsible for creating the user interface for the Information Bridge client component, although a couple of pre-defined user interface components already exist in Information Bridge Framework. The developer must tie the user interface to the actions using Metadata Designer.

Once the developer has created the metadata, it can be published to the metadata database. This is done through the Metadata Web service. The Information Bridge client component accesses the metadata through the Web service to determine which user interface elements to show for which actions, and to make calls to the backend services when an action is requested.

Data and Information Flow

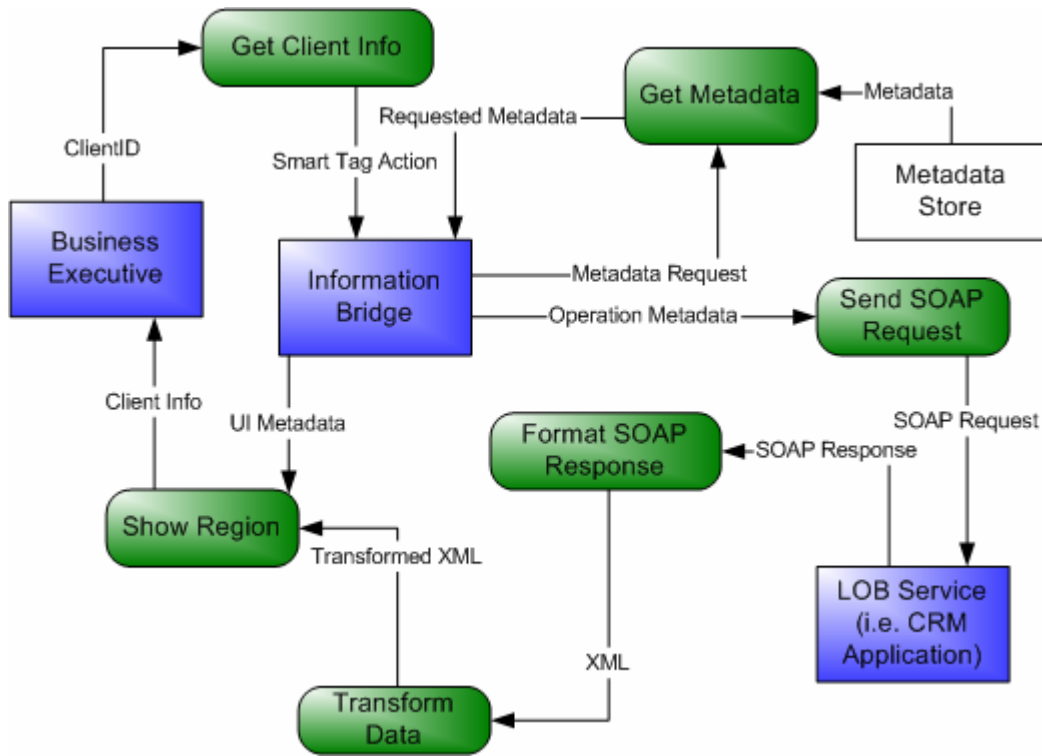
An Information Bridge request usually begins from recognized or “tagged” text within an Office document. However, clicking a specially crafted hyperlink in Internet Explorer can also initiate an Information Bridge request. For example, assume that Smart Tags are created for all of BlueVision’s clients and that those tags are recognized by Outlook. Now assume that a corporate executive receives an e-mail message from a client regarding an unsatisfactory service, and that the client is not considering a contract renewal. The executive would want to access more information about that client to determine whether to look further into satisfying that client or whether the contract renewal is not worth the effort.

The following Level 0 data flow diagram (DFD) illustrates how data and information flow through an Information Bridge solution for this request:



In the previous diagram, the business executive selects the client identification number from within the e-mail message and chooses the “GetClientInfo” Smart Tag action. The action then executes a SOAP call to a Web service that wraps the backend LOB service, which could be a CRM application. The Web service then returns the client information in the SOAP response to Outlook. The business executive can then quickly make an informed decision.

In actuality, there is a lot more going on behind the scenes. The previous diagram shows only the highest level data and information flow. The following diagram breaks the process into several subprocesses, and it provides a more detailed view of what is going on:



In this diagram, the process of retrieving client information is further refined. Specifically, six subprocesses are involved. The “Get Client Info” process uses an Information Bridge Smart Tag action to request an operation to be performed against the LOB service. Information Bridge must first access the metadata associated with the action from the metadata database by means of the Metadata Web service. When the requested metadata is returned, Information Bridge uses it to send a SOAP request to the LOB service. (Remember, the metadata contains instructions on how to access backend services, as well as how to show the user interface for viewing the requested data.) When the SOAP response is returned, it can be formatted using XLST if formatting has been specified in the metadata. That data is then sent to the “Show

Region” process. The Visual Studio .NET developer must know how to navigate through this data from within the user interface since the data is represented as XML. The user interface is then presented to the business executive.

Understanding Information Bridge Metadata

The centerpiece of Information Bridge solutions is metadata. Metadata can be divided into two types: service metadata and solution metadata. Service metadata describes how Information Bridge should connect to backend LOB services (for example, which SOAP operations to call). Solution metadata describes how Information Bridge should use the data retrieved from LOB services to display the user interface to the end user.

The Metadata Hierarchy

The following metadata hierarchy lists the metadata elements that you will likely use in your Information Bridge solutions.

```

Metadata
  MetadataScopes
    MetadataScope
      Entities
        Entity
          Views
            View
              MenuItemDefinition
                IsVisibleOperationInstance
                IsEnabledOperationInstance
                MenuDefinitionOperationInstance
              PortInstances
                PortInstance
                  AuthenticationDefinitionOperationInstance
                  LocationOperationInstance
            Actions
              Action
                MenuItemDefinition
                  IsVisibleOperationInstance
                  IsEnabledOperationInstance
                  MenuDefinitionOperationInstance
                  PortInstances
    
```

```

    PortInstance
      AuthenticationDefinitionOperationInstance
      LocationOperationInstance
  OperationInstances
    OperationInstance
      TransformationInstances
      TransformationInstance
  PortInstances
    PortInstance
      AuthenticationDefinitionOperationInstance
      LocationOperationInstance
  Relationships
    Relationship
      MenuItemDefinition
      IsVisibleOperationInstance
      IsEnabledOperationInstance
      MenuDefinitionOperationInstance
      PortInstances
        PortInstance
          AuthenticationDefinitionOperationInstance
          LocationOperationInstance
  ViewLocators
    ViewLocator
      OperationInstance
      TransformationInstances
      TransformationInstance
      PortInstance
      AuthenticationDefinitionOperationInstance
      LocationOperationInstance

  Groups
    Group
  Operations
    Operation
      Groups
        Group
      InputSchemas
        SchemaInstance
      OutputSchemas
        SchemaInstance
      Properties
  Ports
    Port
  Schemas
    Schema
  Transformations
    Transformation
  Translations
    Translation
  MetadataSolutions
    MetadataSolution
  OperationTypes
    OperationType

```

Metadata Scopes

Metadata scopes provide the logical separation of LOB applications or areas of functionality. For example, you could create one scope for the human resources department, another scope for the support department, and a third for the accounting department. Each scope can have its own entities, ports, operations, transformations, schemas, translations, and groups, as indicated by the hierarchy.

Entities

Entities are the equivalent of conceptual business objects within an organization. They typically represent people, places, or things that are used in the context of LOB applications. Entities are meant to represent high-level business objects, while views are designed to represent a particular data structure of an entity. For example, the human resources department might view salary information for an employee, while the support department might only view the employee's title and job queue. However, both departments would be viewing information on an employee, the entity. A single entity could therefore be constructed of data from multiple LOB applications.

Ports

Ports specify the locations or contents of data sources, such as XML files, XSLT files, .NET assemblies, and Web service endpoints. Operations, schemas, and transformations are sometimes mapped to ports.

Schemas

Schemas define the data structures that are used in an Information Bridge solution. This includes data structures for both LOB data and Information Bridge data. Schemas define the

input and output parameters for operations. When schema validation is turned on, schemas also validate data that is passed to and returned from operations. Each schema is mapped to a port that contains the actual data that defines the schema.

Operations

Operations represent the individual and atomic tasks that can occur as part of a particular action. For example, if an executive wants to view information about a client, that executive would execute a single action, which could consist of the following operations in sequence:

1. Invoke a Web method on a Web service to retrieve data.
2. Display the returned data in the Information Bridge information window.

Whenever an operation exists as part of an action, that operation is actually considered an operation instance. In other words, operations themselves are basically abstract. Whereas an operation defines the input and output schemas for the parameters and return value, an operation instance can define the actual data that is to be passed as input to the operation, as well as any transformation to be used on the data. The input data is usually view data, reference data, action data, or output from another operation, as you will learn in later chapters.

Transformations

Transformations allow data to be transformed before it is used. Some types of transformations include regular expressions, XPath, and XSLT. Similar to operations, transformations by themselves are abstract. Transformation instances exist when transformations are applied to operation instances. Whereas a transformation defines the input and output schemas affected by the transformation, transformation instances define the actual data to be transformed. A transformation instance can be applied to an operation instance for each input parameter that the operation accepts.

Translations

Translations allow you to translate data from one language to another. This allows you to localize LOB data.

Groups

Groups allow you to assign permissions to operations. They exist as task definitions in Authorization Manager. A security engineer can then assign these tasks to roles, and grant permissions on the roles to Windows user and group accounts.

Operation Types

Every operation that you define must be associated with an operation type. This is because Information Bridge is responsible for invoking the operations, and therefore it must know how to do so. The operation types supported by Information Bridge are defined next. However, you might not gain a complete understanding of them until you actually use them.

OperationBrowse

Operations of this type allow you to retrieve data from HTTP locations. For example, you can load data from an XML file, a text file, or a binary file.

OperationCallComponent

Operations of this type can invoke class-level methods in .NET assemblies. Information Bridge uses CLR reflection to invoke these methods. For these operation, you must specify the port that contains the assembly location, the class that contains the method, and the name of the method to invoke. The parameters and return value must be serializable.

OperationExecuteAction

Operations of this type can execute actions that are defined in metadata. For these operations, you must specify the target scope, entity, view, reference schema, and action name. The reference schema determines how to retrieve the view, as you will learn later.

OperationNavigate

Operations of this type allow you to navigate from one view to another using a relationship that is defined in the metadata. Relationships describe how views are associated with each other.

OperationQueryMenu

Operations of this type retrieve menu definitions for actions, views, and relationships. Every menu item definition is assigned a menu type. An OperationQueryMenu operation is also assigned a menu type. The menu definitions that it retrieves are those that are assigned the same menu type as the operation.

OperationReturnActionResult

Operations of this type return the results of actions that have been executed through an OperationExecuteAction operation.

OperationSoapRequest

Operations of this type invoke Web methods on Web services. For these operations, you must specify the port that defines the SOAP endpoint, as well as the SOAP action that identifies the Web method to invoke.

OperationStatusCheck

Operations of this type are typically used in conjunction with OperationQueryMenu operations to determine whether menu items should be visible and enabled. An OperationStatusCheck operation is basically an OperationCallComponent operation that returns a *Boolean* value.

OperationTransformationAggregation

Operations of this type allow you to transform data from one or more input schemas into a single output schema. These operations use XSL transformations to transform the data.

OperationCustom

This is a custom operation type, which means that it is not necessarily implemented by Information Bridge. However, Information Bridge Framework does include a few custom operation types. Every custom operation type accepts zero or one input and output schemas, as well as an optional XML node that represents the properties of the operation. You should create custom operation types when built-in types do not suffice.

Planning Information Bridge Framework Solutions

One of the most common scenarios in which an Information Bridge solution would be useful is when users are required to access multiple LOB applications to perform a single task or make an informed decision. Information Bridge allows you to group these multiple operations into a single action, which can be executed from within a single Office application.

Another common scenario is when a user needs to retrieve information about a business entity that is represented in a document. For example, when an Information Bridge solution is in

place, a lawyer need not access the firm's legal claims application when viewing an Excel spreadsheet of new cases to quickly determine which types of cases have been won or lost. Information Bridge would allow the lawyer to quickly accept or reject a specific type of case.

Once you have determined the need for an Information Bridge solution, you need to take time to plan its implementation. The basic concept to remember is to start with a small solution, but make it flexible enough to grow. Because Information Bridge is not designed to replace LOB applications, you should not attempt to implement a solution that solves every business need. Start with a small solution that makes a big impact.

Entity Planning

Entity planning consists of determining the business entities (persons, places, or things) that will be used in the Information Bridge solution. Think of the most basic entities first, because they can be refined in future solutions. The business analyst is usually responsible for this stage of planning.

Web Services Planning

Web services planning consists of determining the backend operations that should be exposed as Web methods. You should design Web services in a way that they will be used to access LOB data from any client, not just from Information Bridge. This way, Web services can be reused in multiple Information Bridge solutions, as well as in other types of solutions. The software architect is responsible for Web services planning. In general, a Web service should map to a class of LOB operations, while each LOB operation should map to a Web method.

Metadata Planning

Metadata planning can be divided into service metadata planning and solution metadata planning. Service metadata planning consists of describing the metadata that will be used to access one or more Web services. Solution metadata planning consists of describing the metadata that will be used to display information that is returned from an LOB service. There is a very thin line between metadata planning and metadata implementation. So don't feel compelled to follow any specific software model when planning and implementing metadata.

User Interface Planning

User interface planning consists of determining how end users will interact with your Information Bridge solution, how the display should be structured, and how the user interface should meet accessibility requirements. The developer or user interface designer is responsible for user interface planning.

Security Planning

Security planning consists of determining which users will need to access certain LOB data. Security needs to be incorporated into Web services planning, metadata planning, and user interface planning. For example, you might want to display only relevant information for users who are members of a specific role. You could choose to develop a user interface for each role, or you could simply show or hide content in a single user interface.

With regards to Web services, you need to control authorization to the Web methods. With Information Bridge solutions, this is usually done using Authorization Manager. However, you might want to implement additional security by securing each Web method. The reason that you cannot rely only on Web services security is because it cannot be used to dynamically grant or deny access to Information Bridge operations. Either a user is granted access to call a

Web method (which might be mapped to an Information Bridge operation), or the user is denied access. However, an Information Bridge operation instance might exist in multiple actions. The user might have permission to execute one of those actions, but not the others. Authorization Manager makes this possible. When a user initiates an Information Bridge request to the Metadata Web service, the Web service looks up the user's credentials in Authorization Manager to determine if that user has been granted access to the operation indicated by the requested metadata. If not, the user will not be able to execute the operation for that request. The credentials that are passed from the Information Bridge client component to the Metadata Web service are always those of the currently-logged on user. In a traditional Web services client application, you could implement the client application to pass the credentials of the logged-on user to the Web service, which could authenticate the user in Active Directory or some other credentials store. You cannot easily tap into Information Bridge to perform this same task. However, you can dynamically set the credentials that are passed from the Information Bridge client component to any backend LOB services (just not the Metadata Web service). For example, you can use Web services security to secure a backend LOB Web method to ensure that a particular user never has permission to execute that method. This effectively overrides Authorization Manager security.

Deployment Planning

Deployment planning consists of determining where the Metadata Web service will be deployed, determining scalability requirements of the LOB Web services, and figuring out a scheme to deploy the solution to Office users within the organization. Deployment is usually performed by an information technology (IT) engineer or administrator.

Installing Information Bridge Framework

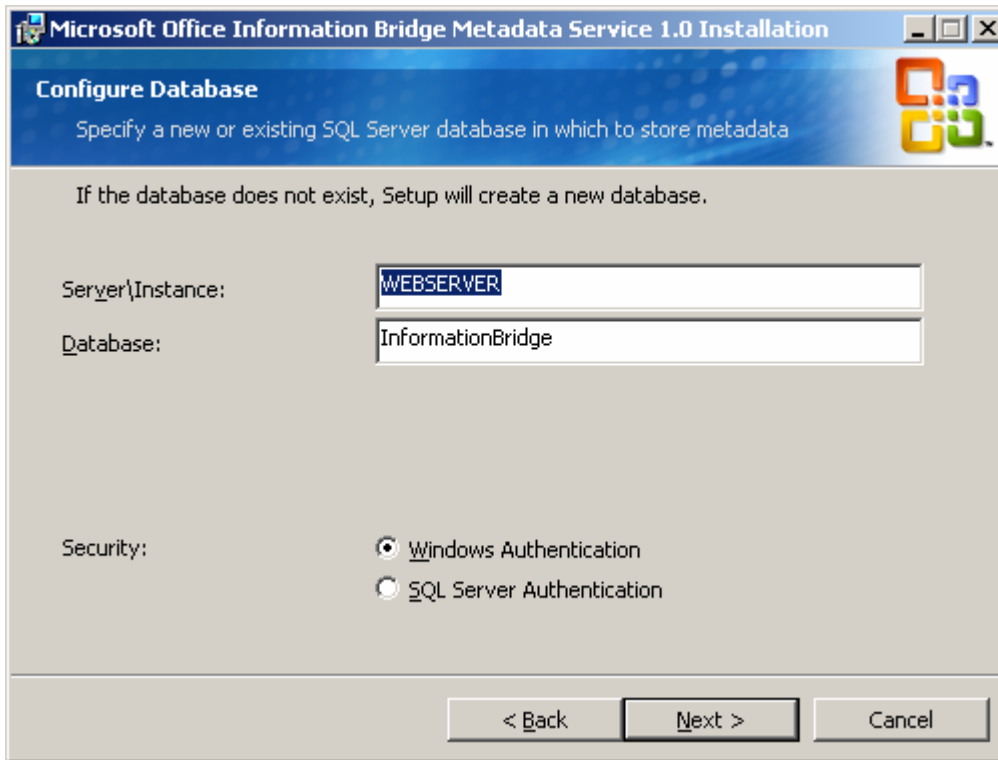
Installing Information Bridge Framework requires running three separate installations: the client installation, the development installation, and the server installation. I will go over each of these in detail.

Server Installation

You should begin server installation by designating a server computer running Windows Server 2003 and SQL Server 2000 Service Pack 3 to host the Metadata Web service. Actually, you could install SQL Server 2000 on a separate computer, but you would need to configure SQL Server security to allow the Web service to access the metadata database, which is used as the metadata data store.

Once you designate a server computer, you need to allocate two Hypertext Transfer Protocol (HTTP) ports that can be used by Information Bridge to access the Metadata Web service. The Metadata Web service actually consists of two endpoints: a read service named `IBFReadService.asmx` and a write service named `IBFWriteService.asmx`. By default, the server installation program uses ports 8081 and 8082 for the read and write services, respectively. Note that you do not necessarily need to specify any ports at all. These ports simply allow you to differentiate HTTP requests. Port 80 is the default HTTP port, and it is usually used to serve both internal and external clients, but the Metadata Web service is typically called from within an organization. Specifying ports for the Metadata Web service allows you to configure any firewalls accordingly.

Run the `Microsoft.InformationBridge.Framework.Server1.0.msi` setup program on the server computer to invoke the installer. Click Next on the Welcome page and then accept the terms of the license agreement. When you click Next, you will be presented with a page allowing you to configure the database, as shown here:



On this page, you should specify the SQL Server instance and database name that will host the metadata. If you configured named instances rather than the default instance of SQL Server, be sure to specify the named instance (for example, WebServer\Production). You should also choose Windows authentication for increased security, although SQL Server authentication is acceptable.

If you choose Windows authentication, the installation program will grant appropriate database access to the Network Service account. This is the default account under which the Metadata Web service will run. If you host additional Web services and Web applications on the same Web server that also run under the security context of Network Service, you might want to change the account under which the Metadata Web service runs. Otherwise, those applications and Web services would be able to directly access and manipulate metadata in the database. If

you do change the account under which the Metadata Web service runs, be sure to change the database security settings to reflect the new account.

If you choose SQL Server authentication, you will need to specify a user name and password that the Web service will use to access the metadata. This account information will be stored in the Web.config file for the Web service. This is why SQL Server authentication is less secure than Windows authentication. Also, you cannot enforce password policies with SQL Server security.

When you click Next, you can choose whether to install only the read service, only the write service, or both. By default, both services are selected to be installed. You can also choose the location for the Information Bridge authorization store that will be used with Authorization Manager. You should specify a network share if you plan to host the two metadata services on separate Web servers.

Next, you need to specify the read and write ports, which default to 8081 and 8082, respectively. For development and testing purposes, it is easier to keep the default settings. However, in a production environment, you might want to change these port settings as required by your company's firewall and security policy. When you click Next, installation will proceed. Note that the account under which you launch the installation process must have permission to create the necessary database objects in SQL Server.

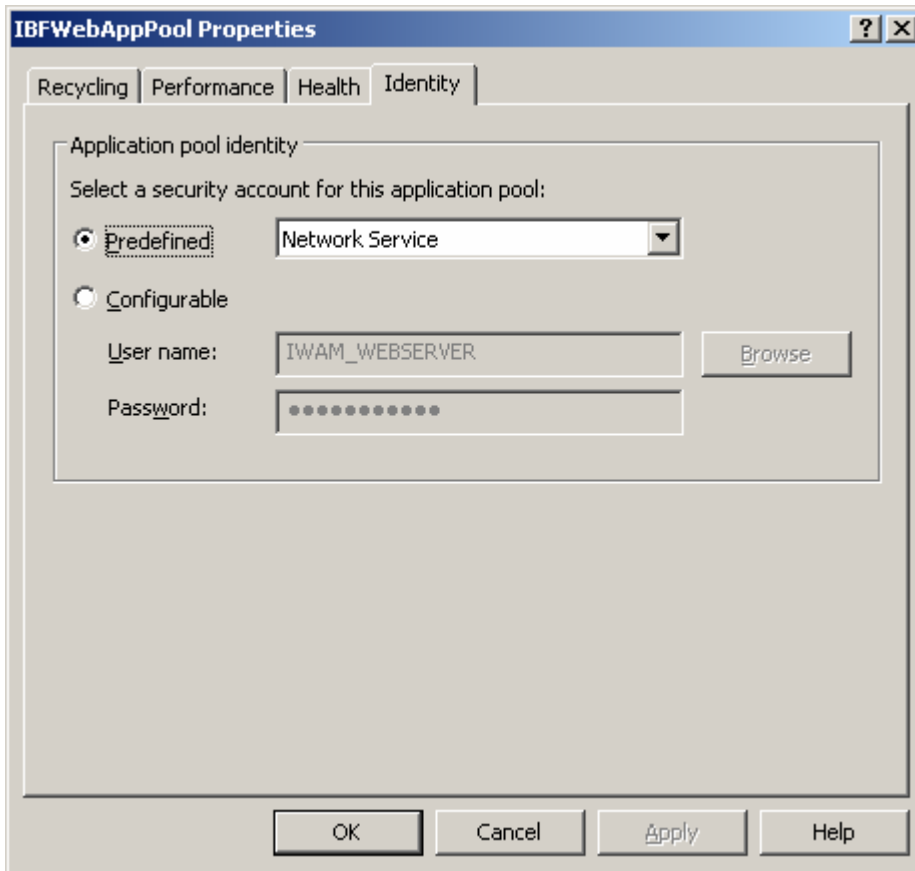
After installation is complete, the following folders will exist in the Inetpub folder on the system drive: IBFReadService and IBFWriteService. Additionally, if you used the default location for the authorization store, a folder named azRoles will exist. The IBFReadService and IBFWriteService folders contain the assemblies and configuration files for the read and write services, respectively. To test connectivity to the services, make the following change in the Web.config file for each service:

```
<webServices>  
  <protocols>
```

```
<!--<remove name="Documentation"/>>  
</protocols>  
</webServices>
```

In the previous configuration, a protocol named “Documentation” is removed from the Web service. By default, the Web service is configured to prevent you from viewing the Web Services Description Language (WSDL) documentation that describes the Web service. If you now type the URL for the services in your browser (for example, <http://localhost:8081/IBFReadService.asmx>), you should see the WSDL documentation. Now undo the change that you just made.

If you open IIS and expand the Application Pools folder for the Web server, you should notice a new application pool named IBFWebAppPool. An application pool allows multiple Web applications and Web services to run under the same account and worker process, while separating them from other applications and Web services in different application pools. This way, a critical failure of a Web application in one application pool cannot cause a Web application in another application pool to fail. Both the read and write services run in the IBFWebAppPool. If you open the application pool’s Properties dialog box and click the Identity tab, you will notice the name of the identity under which the read and write services run, as shown here:



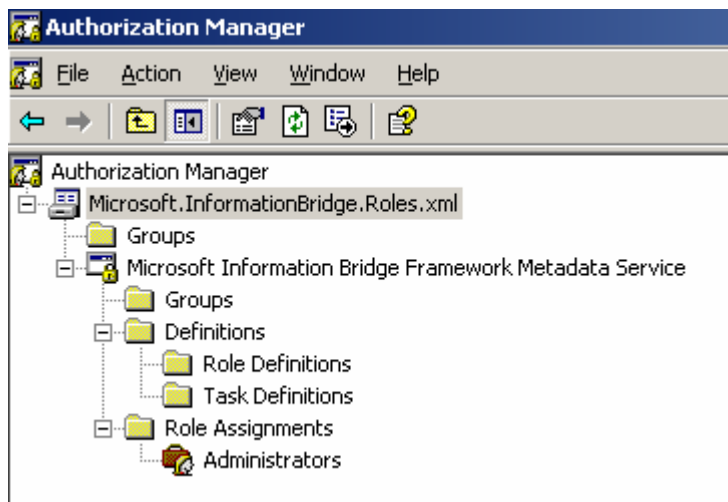
You can change the identity to any Windows account. However, the account you choose will need permission to access the metadata from the SQL Server database.

The read and write services are both configured to use integrated Windows authentication, and this should not be changed. The Metadata Web service uses Authorization Manager to grant access to operations, and this can only be accomplished with Windows user accounts.

If you open SQL Server 2000 Enterprise Manager, expand the InformationBridge database (or whatever database name you chose) for the SQL Server instance that you specified, and click the Users node, you should see a database user named NetworkService. Open the Properties dialog box for this user account, and you should notice that it is a member of three database

roles: public, iw_b_MetadataReader, and iw_b_MetadataWriter. The iw_b_MetadataReader role is granted permission to execute all metadata-related stored procedures that retrieve metadata from the database; the iw_b_MetadataWriter role is granted permission to execute all metadata-related stored procedures that insert, update, and delete metadata. If you change the identity under which the Metadata Web service runs, and you want the Metadata Web service to use Windows authentication to access SQL Server, you must remember to add the new account to these two roles. If you install the read and write services on different Web servers and/or with different identities, you must add each identity to the appropriate database role.

As stated previously, Information Bridge uses Authorization Manager to grant access to metadata. If you open the folder that represents the location for the authorization store (for example, C:\Inetpub\azRoles), you should notice an XML file named Microsoft.InformationBridge.Roles.xml. This file contains the name of the authorization store, applications, roles, tasks, and operations that are used by Information Bridge. The first thing that you need to do is import this file into Authorization Manager. On the Web server, click Start, Run, and then enter Azman.msc in the Open dialog box. This launches the Authorization Manager Microsoft Management Console (MMC) snap-in, as shown here:

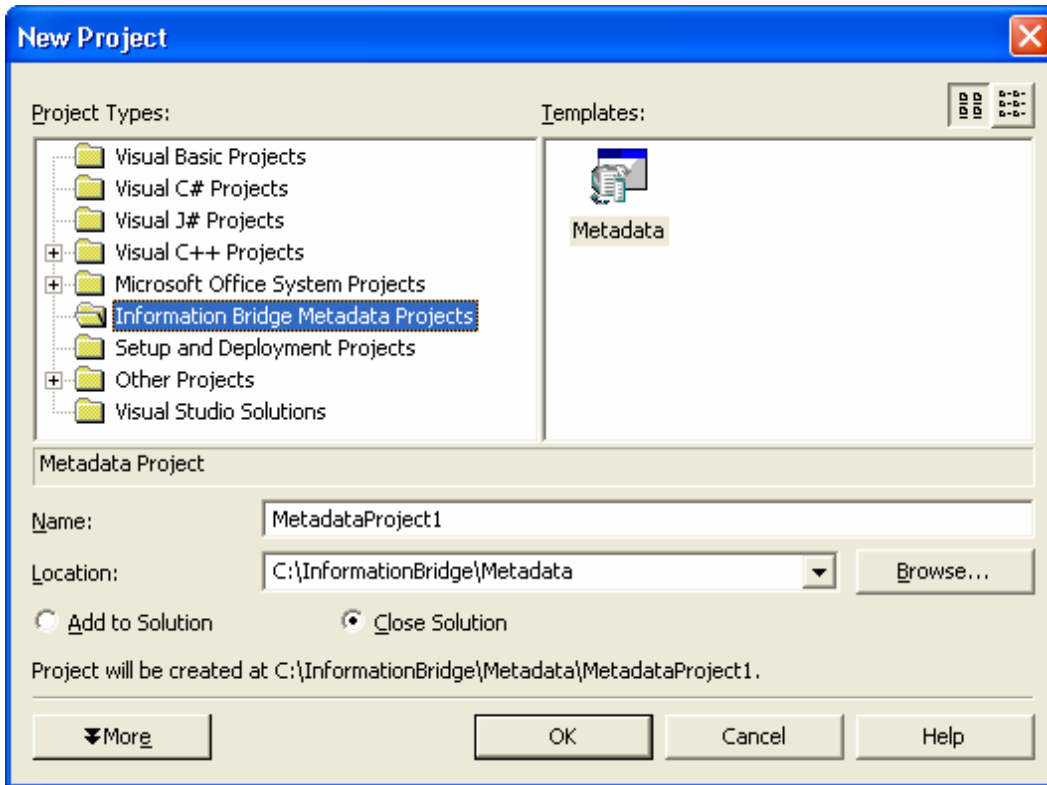


With Authorziation Manager, you can grant an application-specific role permission to perform a task within the application. A task consists of one or more operations or lower-level tasks. However, you cannot define operations within Authorization Manager unless you switch to Developer mode. You can do this by choosing Options on the context menu of the Authorization Manager node.

You can also define application scopes in Authorization Manager. An application scope simply allows you to group tasks that share the same role requirements. For example, you could use scopes to allow an employee to update a client's information using a CRM application, but prevent that same employee from updating a client's information using a sales application. Although updating the client's information consists of the same task in both applications, the task exists in different business areas or "scopes." In the context of Information Bridge, only one logical application exists, which is the Office application. Therefore, you need to use scopes to further refine access requirements.

Development Installation

After you have installed the server components, you can install the development components by running the Microsoft.InformationBridge.MetadataDesigner1.5 msi setup program on the development computer. This installer creates a new project template and the Metadata Designer tool for use with Visual Studio .NET. Accept the terms of the license agreement and specify the installation path to install the metadata tools when requested. When installation completes, you can open Visual Studio .NET and you should notice a new project type, as shown here:

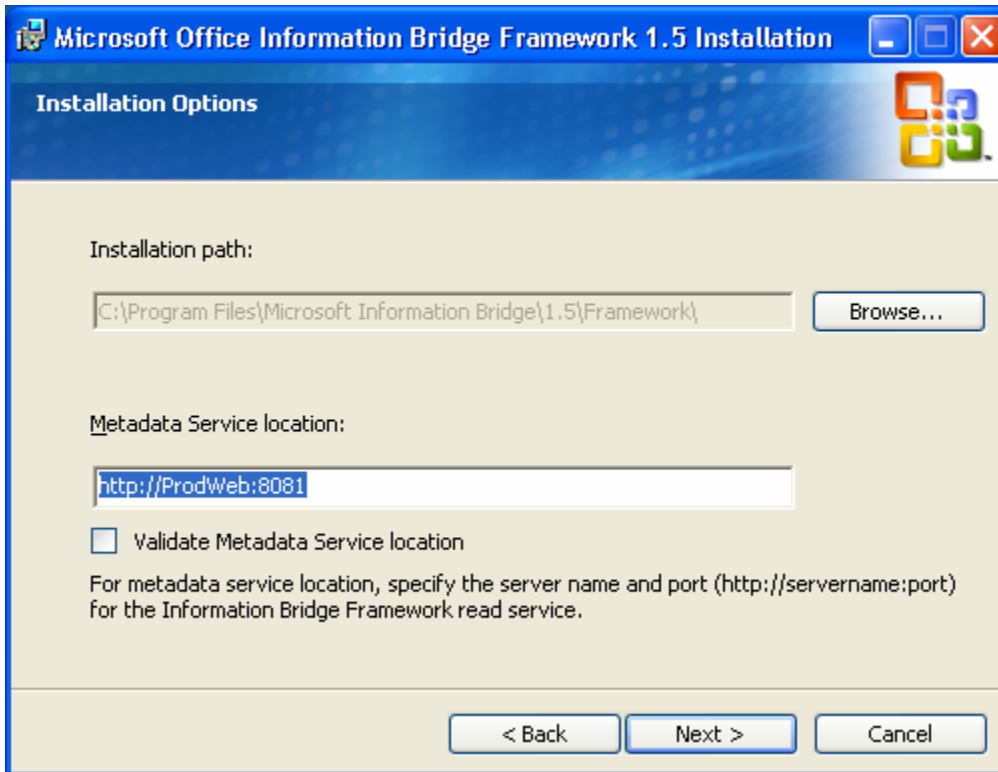


When you create a new metadata project, you can use Visual Studio .NET to design metadata and publish it to the Metadata Web service, or you can use it to access existing metadata from the Metadata Web service and modify it.

Client Installation

Finally, you should install the client components on either your development computer during development and testing or on a user's computer when rolling out your first Information Bridge solution to production. Ideally, the Information Bridge client component will be installed prior to your Information Bridge solution deployment in a production environment.

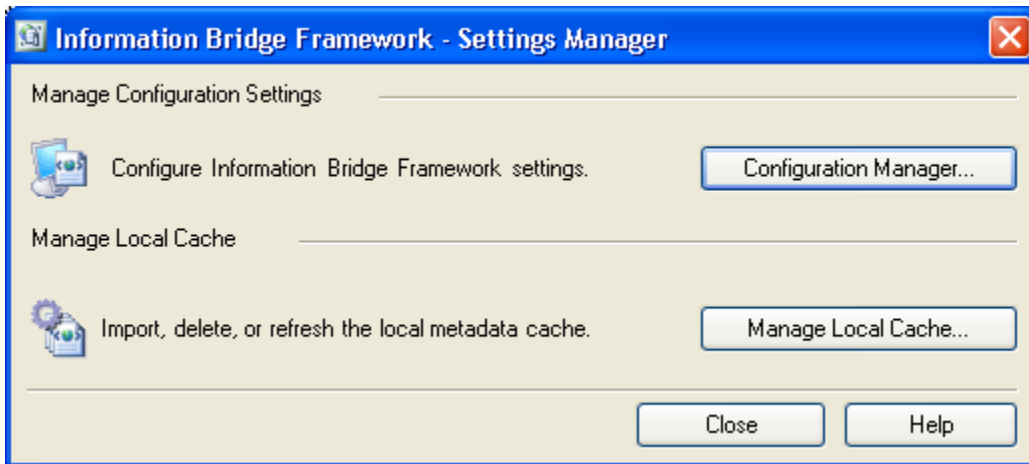
To start the client installation process, run the Microsoft.InformationBridge.Framework.Client1.0.msi setup program. After you accept the terms of the license agreement, you will be presented with a page that asks you for the installation path and the Metadata Service location, as shown here:



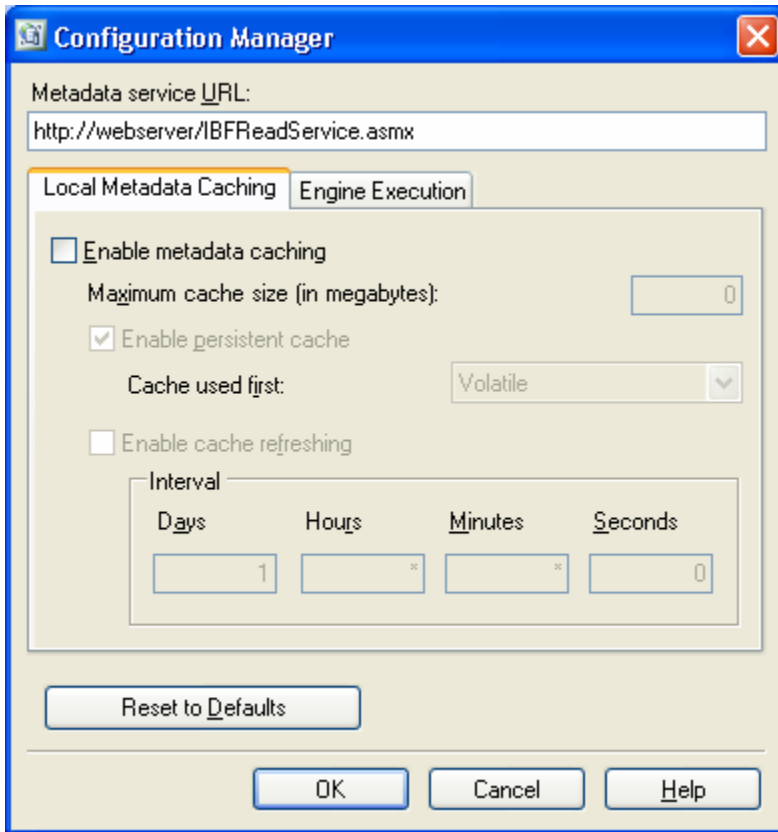
On this page, you need to specify *only* the server name and port to access the read service. For example, if you installed the read service on a Web server named ProdWeb with the default port, you need to specify the following: http://ProdWeb:8081. (Do not specify the name of the read service, i.e. IBFReadService.asmx.) You should check the Validate Metadata Service location option to verify that the location you entered is valid.

As of this writing, I have not been able to get the installation program to validate the location successfully when I choose this option, even though the Information Bridge client components access the read service correctly after installation. If you receive an error message, uncheck the option.

After installation completes, you can verify that the correct location was configured by viewing the `Microsoft.InformationBridge.Framework.config` file in the installation directory. Examine the `Url` attribute of the `MetadataService` element. If it does not specify the correct location to the read service, change it. The full path to the read service should be specified, including `IBFReadService.asmx`. You can also check this by opening the Information Bridge Settings Manager Control Panel applet in the Administrative Tools folder, which is shown here:



After the applet opens, click the Configuration Manager button to open the Configuration Manager dialog box, which is shown here:



Installing Information Bridge Framework Resource Kit

Information Bridge includes a resource kit that can be downloaded separately from the Information Bridge Framework installation. It can be downloaded at the following URL:

<http://msdn.microsoft.com/office/understanding/ibframework>

The resource kit consists of a set of tools, documentation, and sample solutions that are designed to help you develop Information Bridge Framework solutions. The following table lists the available tools and a description of each:

Tools	Description
Metadata Publisher	This tool allows you to import and publish metadata without having to open Visual Studio .NET.
Group Policy ADM File	This file contains settings that you can import into Group Policy so that you can configure all Information Bridge client computers from a central place.
Visio Metadata Explorer	This tool allows you to use Microsoft Office Visio to design or view metadata for Information Bridge solutions.
WSE 2.0 Adapter	This tool allows you to incorporate WS-Security into your Information Bridge solutions. WSE 2.0 must be installed before you can use this tool.

In addition to the tools listed in the table, the resource kit also includes metadata templates that you can import into your custom Information Bridge solutions; a sample solution that integrates with Excel, Outlook, Word, InfoPath, Sharepoint, and SQL Server Reporting service; and a solution development guide aimed to help you create Information Bridge solutions using best practices. This concludes installation of Information Bridge Framework.

Summary

This chapter discussed the basic architecture behind Information Bridge Framework. Information Bridge Framework consists of a client component that will be accessed by Office users, a Metadata Designer that the developer will use to implement metadata, and the Metadata Web service that provides access to the metadata store.

This chapter also discussed how to plan Information Bridge Framework solutions. Planning should begin with the designation of entities, which are simply the business objects (persons, places, and things) used throughout your LOB applications. It also involves planning the development of one or more Web services that will be used by Information Bridge to access and manipulate the LOB data. User interface planning allows you to determine the best way to display that data to the Office user, while deployment planning helps you develop a strategy to get the solution to the user. Security planning must be performed at all of these stages, while metadata planning might not occur at all until implementation.