



Working with BizTalk Adapter Pack 2.0

Technical White Paper

motion10
10



Table of Contents

Background	3
Table 1: Adapters included in 'Microsoft BizTalk Adapters for Enterprise Applications'	4
<i>Figure 1: LOB Adapter process flow</i>	5
Architecture.....	9
Installation and Configuration.....	11
The BizTalk Adapter Pack 2.0 in action.....	17
Scenario and walkthroughs	17
Adapter with BizTalk Server Walkthrough	18
Adapter with .NET Walkthrough	25
Conclusion	28
Resources	28
Acknowledgements.....	29
About motion10.....	29





Microsoft's BizTalk Server offering contains the BizTalk Adapter Pack. The Adapter Pack provides interoperability with multiple Line-Of-Business (LOB) applications like SAP, Siebel or Oracle. Currently version 2.0 of the adapter pack is available supporting BizTalk Server 2006 R2 and BizTalk Server 2009 for Developer, Standard, and Enterprise Editions. The Adapter Pack can be purchased separately and is licensed on a per processor basis and BizTalk Server is not required. BizTalk Adapter Pack 2.0 is a collection of key Line-of-Business (LOB) application adapters that enables any Windows application to integrate with LOB applications using the Windows Communication Foundation (WCF) programming model. These adapters are written using the WCF LOB Adapter SDK, a prerequisite to be able to use the pack. This article will dive into the background of the adapter pack, its architecture, how to install and configure the pack and describes a scenario of how the pack could be used.

Background

As BizTalk Server evolved the number of adapters grew the Enterprise Adapters became first class citizen of BizTalk adapter's collection. Out of the box BizTalk offers a lot of adapters that support numerous databases, protocols and applications. The adapter pack extended the number of adapters to provide more integration capabilities.

Interoperability to SAP has been supported by Microsoft since 2000 with connectivity to SAP R/3. Adapters for SAP were based initially on the BizTalk Server Adapter Framework and later on the Adapter Pack 1.0 release which makes use of the Windows Communication Framework (WCF). Besides SAP Microsoft in August 2005 acquired eight .NET based application adapters from iWay. These adapters allowed BizTalk Server customers to easily connect to third-party applications like Amdocs, JD Edwards, Oracle Corp., PeopleSoft, Siebel Systems Inc. and TIBCO Software Inc. This purchase of the iWay adapters took place before the major release of BizTalk Server (2006), SQL Server (2005) and Visual Studio (2005). These adapters, which were based on the BizTalk Server Adapter Framework (.NET Framework 2.0), were bundled into the "Microsoft BizTalk Adapters for Enterprise Applications" offering which targeted at BizTalk 2006 and later BizTalk Server 2006 R2.

The Adapter Framework is a set of extensible APIs. The APIs enable developers to take advantage of relevant services within BizTalk Server when building adapters and connectors that integrate third-party applications. Therefore also custom adapters could be made using this framework. Table 1 lists the adapters that are included in 'Microsoft BizTalk Adapters for Enterprise Applications'.





Table 1: Adapters included in 'Microsoft BizTalk Adapters for Enterprise Applications'

Adapter	Description	Supported Version
J.D. Edwards Enterprise One		8.10,8.11, and 8.12
J.D. Edwards OneWorld XE		B7.3.3.3 with Service Pack 23, and B7.3.3.4 with Service Pack 23
Oracle Database	Provides access to Oracle Database tables, views, stored procedures, functions, and PL/SQL packages and is based on ODBC.	8i, 9i, and 10g
Oracle E-Business Suite	Provides access to Oracle eBusiness Suite applications and is based on ODBC.	Oracle E-Business Suite Application 11.5.10.2
PeopleSoft Enterprise		People Tools 8.17.02, 8.43, 8.45, 8.46, and 8.48
Siebel eBusiness Applications	Provides access to Siebel eBusiness Applications via Business Objects, Business Components, and Business Services.	Version 6.2.1 with patch 110 or later, Version 7.0.*, Version 7.5.*, Version 7.7, and Version 7.8
TIBCO Enterprise Message Service		TIBCO Enterprise Message Service (EMS) version 4.2
TIBCO Rendezvous		TIBCO Rendezvous version 7.3



The LOB adapters provided through Microsoft BizTalk Adapters for Enterprise Applications consist of several components. Once these adapters are deployed you might find the following processes running in your environment as you use the design-time and run-time components of the adapters:

- **Adapter Framework** – The adapters are built on the BizTalk Server Adapter Framework. The Adapter Framework is a set of extensible APIs that enable developers to take advantage of relevant services of BizTalk Server when building adapters and connectors that integrate third-party applications.
- **Java Runtime** – The Adapter Framework, which is used by the LOB adapters, is dependent upon the Java Runtime (version 1.4.3 or later).
- **Browser Agent** – This process is used by the Adapter Framework and invoked by Visual Studio .NET. It retrieves and formats metadata from the LOB systems into XSD files that the BizTalk Server development environment can then consume and use. After five minutes of inactivity, it automatically shuts down to release any resources, such as memory, that may have been used. This process appears in Task Manager as browsingagent.exe.
- **Runtime Agent** – This process is used by the Adapter Framework and is invoked by the BizTalk Server runtime (BtsNtSvc.exe) to call business functions on the LOB systems. Unlike the Browser Agent, the Runtime Agent does not shut down after a period of inactivity. This process appears in Task Manager as runtimeagent.exe.
- **Connectors, JAR files** – JAR files contain the set of Java classes specific to the LOB system. The Adapter Framework exposes these classes as Java interfaces, which are used to communicate with the BizTalk Server design-time and run-time executables.

The figure below shows the high-level architecture of both the design-time and run-time components and how they interface with the LOB adapters and the LOB system.

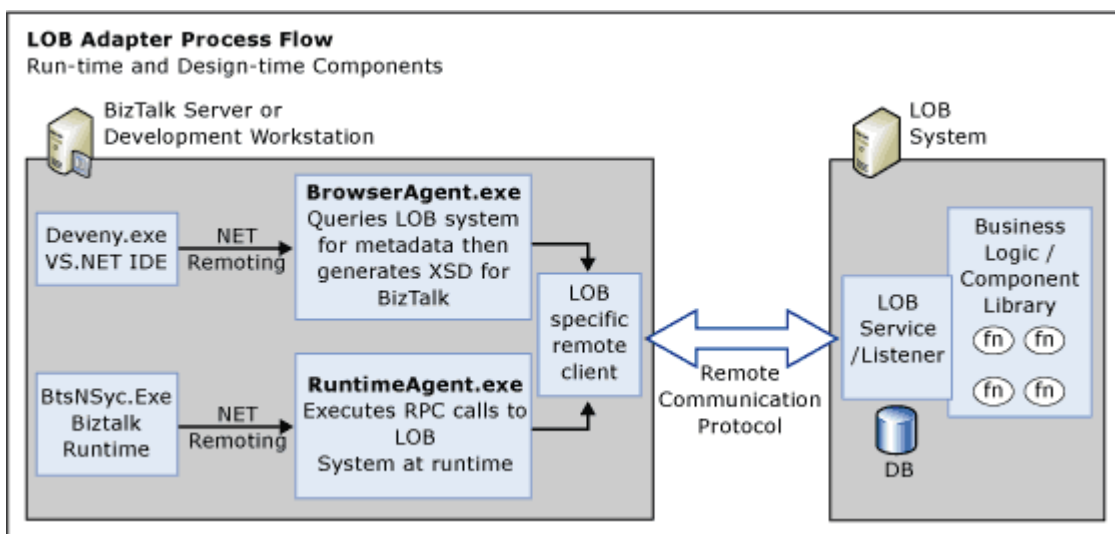


Figure 1: LOB Adapter process flow



Microsoft continued to work on improvements of the LOB adapters and in March 2008 the BizTalk Adapter Pack 1.0 was made available. This pack contained three adapters for Siebel, SAP and Oracle Database based on WCF LOB Adapter SDK. It offered the possibility to connect to a variety of clients (such as Microsoft Office, BizTalk Server, and custom .NET applications). The SDK inherently relies on the core WCF concepts, principles, and classes implemented in the .NET Framework 3.0 and later. It provides for a simplified programming model on top of WCF Channel Model to enable adapter developers in creating service-oriented interfaces to existing systems and applications. Microsoft delivered this Pack to enable a flexible integration and application infrastructure connecting LOB systems with a unified open approach on the technology of the WCF programming model.

The WCF LOB Adapter SDK provides a set of run-time and design-time components, a .NET object model, and support components including:

Component	Description
WCF LOB Adapter Development Wizard	Provides step-by-step guidance in the creation of WCF LOB Adapter SDK projects within Visual Studio .NET.
WCF LOB Adapter Service Development Wizard	Provides step-by-step guidance in creating a Web project to host an adapter in Internet Information Services (IIS).
WCF LOB Adapter SDK run-time system	Supports the WCF LOB Adapter SDK by extending the WCF channel architecture and providing other run-time services.
WCF LOB Adapter SDK object model	A collection of classes, types, and interfaces that support common adapter tasks such as metadata normalization, caching, connection management and pooling, and messaging inspection.
Add Adapter Service Reference Plug-in	Gives custom .NET applications the ability to consume adapters developed using the WCF LOB Adapter SDK.
Consume Adapter Service Add-in	Gives BizTalk Server the ability to consume adapters developed using the WCF LOB Adapter SDK.



The architecture of WCF LOB Adapter SDK will be described in following architecture paragraph. Picture below will give some insight how adapters interfacing with LOB systems changed with BizTalk Adapter Pack.

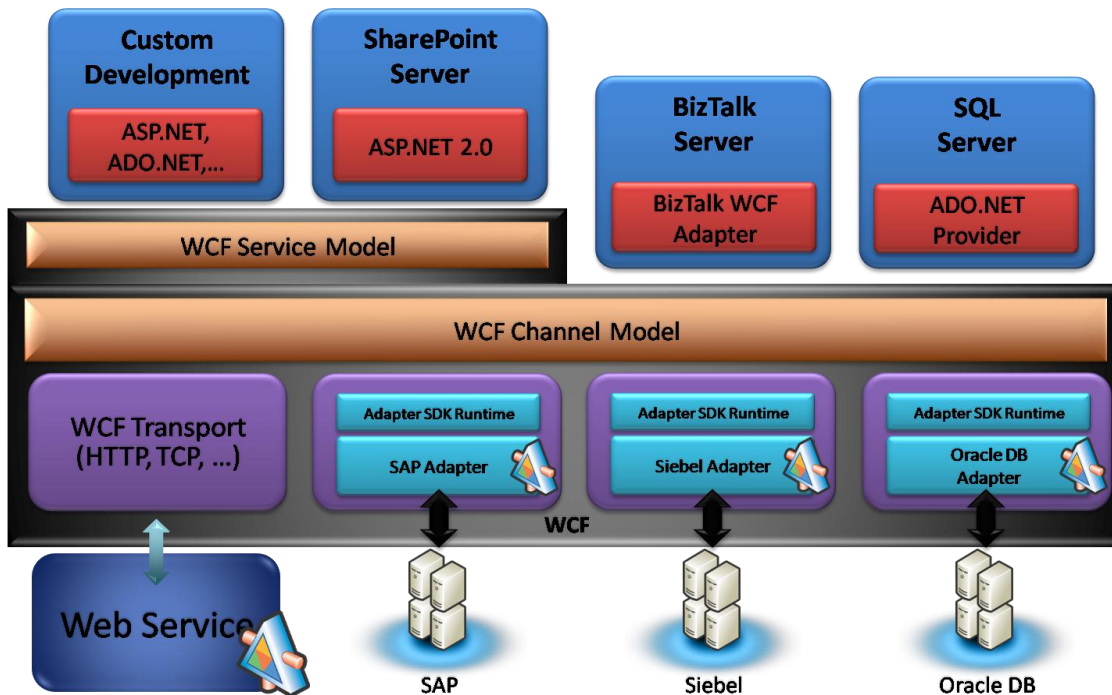


Figure 2: BizTalk Adapter Pack 1.0 Architecture

With BizTalk 2009 Microsoft continued their commitment to simplify interoperability. In Server Adapter Pack 2.0, the BizTalk Server Adapter Pack 1.0 was extended with Oracle E-Business Adapter and SQL Server Adapter. This SQL Server adapter replaces the native SQL adapter offered out-of-the-box from BizTalk Server and offers many more features, to name a few:

- Support for new data types in SQL2005 and SQL2008 (notably FILESTREAM, varbinary (max))

Insert/Update/Delete/Select operations on tables and views

Invoke (SQL and CLR) Stored procedures, scalar valued functions , table value functions

Supports execution of generic T-SQL statements, via the ExecuteReader(), ExecuteNonQuery() and ExecuteScalar() operations

Configurable polling – polling statement, polling interval, receive multiple result sets

Support for x86 and x64 platforms



What is included in the Adapter Pack 2.0 is displayed in table 2 below.

Table 2: Adapter Pack 2.0 features

Adapter	Description	Supported Version
SAP	Supports execution of BAPIs, RFCs, and tRFCs, data exchange using IDOCs, and receiving events from SAP with a generic RFC Server.	SAP R/3 4.x and R/3 6.20 (Enterprise) SAP ECC 6.0
Siebel eBusiness Applications	Provides access to Siebel eBusiness Applications via Business Objects, Business Components, and Business Services.	7.5.3, 7.7, 7.8, and 8.0
Oracle eBusiness Suite	Provides access to Oracle eBusiness Suite applications and is based on ADO.NET.	ODP.NET 11.1.0.7, Oracle EBS 11.5.9, 11.5.10 and 12
Oracle Database	Provides access to Oracle Database tables, views, stored procedures, functions, and PL/SQL packages and is based on ADO.NET. Timed polling is also provided, allowing applications to periodically look for changes in the database.	Oracle 9i (9.2.0.2), 10g (10.1.0.2.0 & 10.2.0.1.0)
SQL Server	Provides access to SQL Server databases.	2000, 2005, and 2008
WCF LOB Adapter SDK	Provides a simple programming model for developing adapters that are based on Windows Communication Foundation. This is a separate download available to everyone using the .NET Framework 3.0.	1.1

What is very important to note here are the supported versions; if for instance you need compatibility with earlier versions of LOB applications, you will need to use older versions of adapters like 'Microsoft BizTalk Adapters for Enterprise Applications' (non-WCF LOB Adapters). These non-WCF LOB adapters are deprecated in BizTalk Server 2009 and to achieve a better supportability you might need to use an older version of BizTalk.



Architecture

Adapters in the Adapter Pack 2.0 are based on WCF LOB Adapter SDK, which consists of a runtime, a collection of APIs, and design-time tools for creating adapters that expose data and operations from line-of-business systems. Adapters manage messages between the adapter consumer and the line-of-business system and can consist of metadata, data, or other information. Figure 3 shows the internal architecture and main components of WCF LOB Adapter SDK..

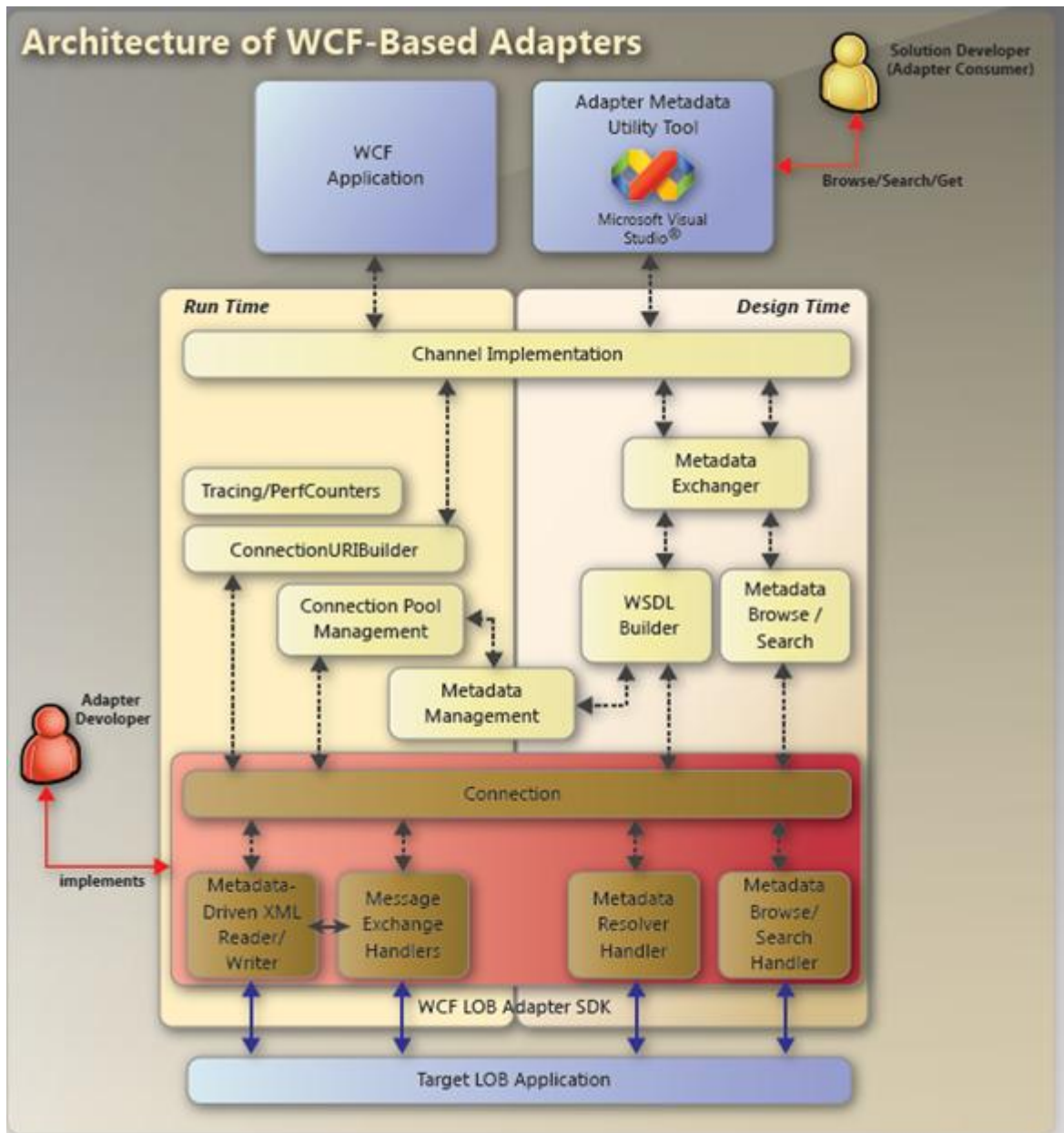


Figure 3: Architecture of WCF-Based Adapters.

Picture above shows two types of developers:

1. Solution developer consuming an adapter.
2. Adapter developer implementing a custom adapter.

Adapter development will not be discussed in this article and focus is on the solution developer working with the Adapter Pack 2.0. Channel implementation is an important part of architecture. An adapter built using the WCF LOB Adapter SDK is basically a transport channel (***System.ServiceModel.Channels.IServiceListener***). It is surfaced to the consumer as a WCF binding, where the binding is used to create the channel stack. This binding can be considered a peer to other predefined WCF bindings such as *BasicHttpBinding*, *WsHttpBinding*, and *NetTcpBinding*, and can be set via app.config or in code by the client application when calling a service. This binding contains an ordered set of binding elements, with the adapter being the key binding element that derives from the ***T:System.ServiceModel.Channels.TransportBindingElement*** class. In an outbound scenario, the WCF LOB Adapter SDK runtime uses a channel factory to create the adapter (that is, the transport channel). This scenario will be used to walkthrough usage of adapter pack 2.0 later on in this article. In an inbound scenario, the WCF LOB Adapter SDK runtime makes use of channel listeners for incoming channels in a service application. Both run-time as well as design-time messages pass through this component.

Other important parts of WCF-based architecture for an adapter consumer are:

- The **Connection Uri Builder** allows adapter consumers to programmatically build connection URIs without specific knowledge of syntax.
- The **Metadata Exchanger** is responsible to handles the client's metadata messages including Adapter SDK's Browse / Search.
- The **WSDL Builder** provides automatic WSDL generation from the WCF LOB Adapter SDK's internal metadata object model (it can be overridden for scenarios that require custom WSDL generation).
- The **Metadata Browse/Search** allows for browsing and searching all LOB metadata.
- The **Metadata management** is responsible for the object-oriented representation of the caching of the metadata for the target system. Metadata can be held in a common cache accessible across all credentials, or it can be cached per credential basis.
- The **Connection Pool Management** is responsible for the lifetime management of the connections of an adapter. It internally keeps a pool of the connections ready for use. This connection pool is credential and URI-based. The credential contains a user name and password that define the security context the connection runs under.

And important tool for solution developers is the **Adapter Metadata Utility** (through Add Adapter Service Reference Plug-in, Consume Adapter Service Add-in) available within Visual Studio, once the WCF LOB Adapter SDK is installed in your development environment. This will be discussed later on in the next paragraph. This utility allows the solution developer to

generating code for the client (for outbound scenarios) and for the service (for inbound scenarios) based on operations selected by the solution developer.

The goal of the WCF LOB Adapter SDK is to facilitate uniform development of reusable metadata-oriented WCF-based adapters that enable enterprise applications, databases and messaging platforms to integrate with each other. The picture below will show this with BizTalk Adapter Pack 2.0 in mind which differs with Adapter Pack 1.0 (see picture 2).

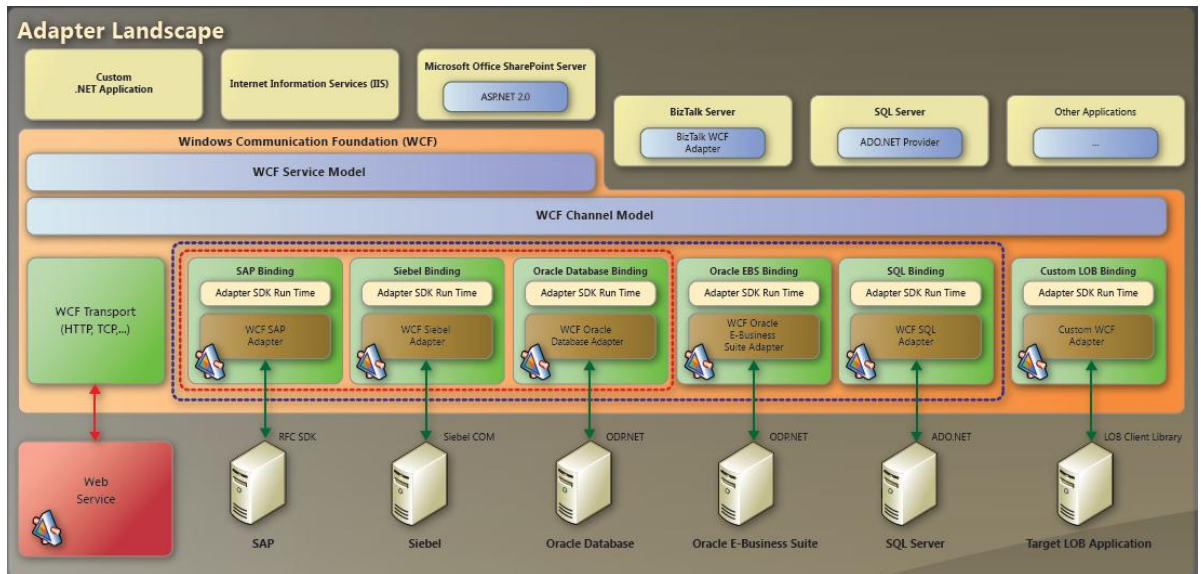


Figure 3: Adapter Landscape of the BizTalk Adapter Pack 2.0.

Installation and Configuration

To install the BizTalk Adapter Pack 2.0 you will need an environment or a virtual machine with for instance the following components installed:

- Operating System like Windows Server 2008 Standard or Enterprise;
- SQL Server 2008 Developer Edition;
- Microsoft .NET Framework 3.5 SP1;
- Microsoft Visual Studio 2008 SP1;
- BizTalk Server 2009.

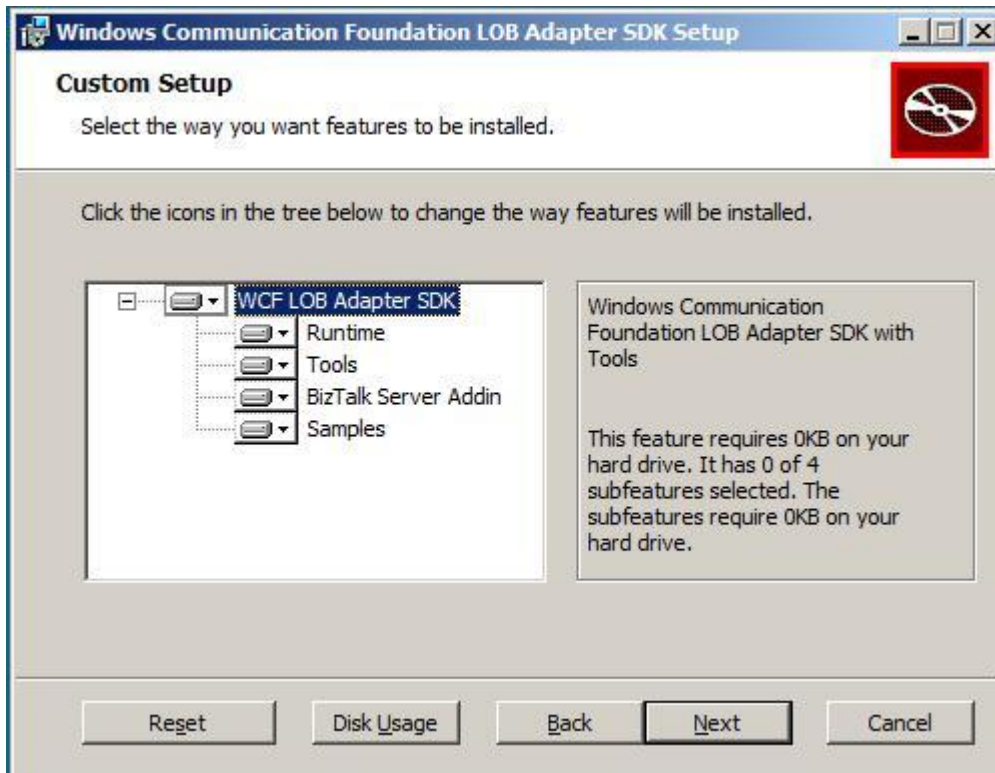
The required environment above will be used as a starting point for usage scenarios for BizTalk Adapter Pack 2.0. Other environments are possible having BizTalk Server 2006R2 and Visual Studio 2005. To install the adapter pack 2.0 you will first need to install WCF LOB Adapter SDK SP2. Installation of the SDK is straight forward, but you may run into a pop window telling you to wait for installer to determine disk space. You can get around this problem by executing the following line (run the installation with reduced UI):

```
msiexec /i "<directoryname>\AdapterFramework.msi"/qr
```

An important note here is that you have to assure that with WCF LOB Adapter SDK you install Runtime, Tools, BizTalk Server Add in and Samples. You can do it running following command line:

msiexec /i "<directoryname>\AdapterFramework.msi"/

You probably will not be bothered by pop-window and you can choose/change and add Runtime, and other components as you can see below in the picture.



Picture 4: WCF LOB Adapter SDK Setup features to install.

The last step to get the Adapter Pack 2.0 installed is the pack itself, available as a 120-day evaluation edition or an officially licensed one through Microsoft. Again a pop-window can ask you to wait for installer to determine disk space and you can use command line if this takes too much time as described above.

Installation of Adapter Pack 2.0 is one thing the other will be an enterprise application supported by the pack (see figure 3). To support the walkthroughs with Adapter Pack and Oracle in coming paragraphs I will shortly describe the installation of Oracle 10g Expression that easily be installed on VM having described configuration.

Oracle Express Edition can be easily obtained through Oracle Technology Network (OTN) after you have registered yourself. For Oracle Database 10g Express Edition installation you can do the following:

1. Password for both SYS and SYSTEM database you can set for instance:
welcome1
2. Below will be shown in next screen and you can leave it as is:
Destination Folder: C:\oraclexe |
Port for 'Oracle Database Listener': 1521
Port for 'Oracle Services for Microsoft Transaction Server': 2030
Port for HTTP Listener: 8080

The above is basically all you have to do for the installation of Oracle 10g Express Edition. The installation will finish with the option to launch the database home page, which you can do to log into the database through SYSTEM account. You then need to unlock the sample user necessary for this scenario. To unlock the sample user account for HR Schema you need to do the following:

1. Click the Administration icon, and then click Database Users.
2. Click the HR schema icon to display the user information for HR.

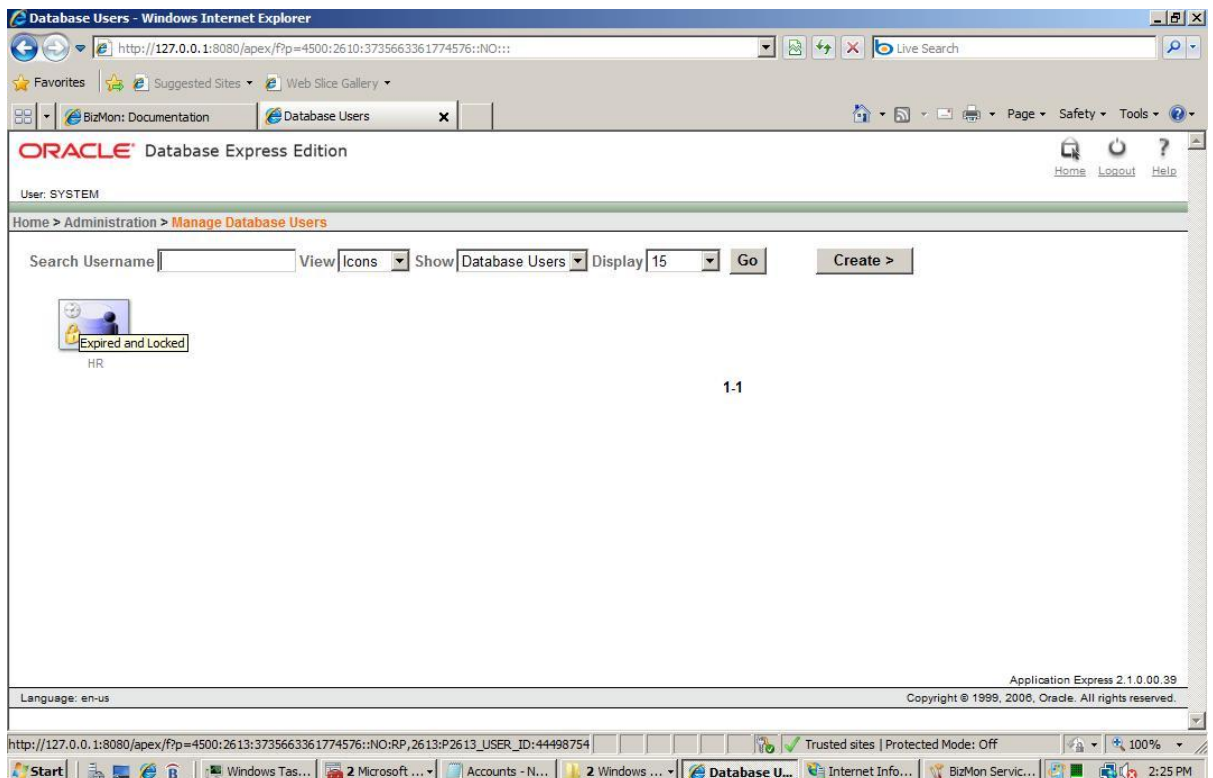


Figure 5: Screenshot of Oracle Database Express Edition: Manage Database Users.

3. Under Manage Database User, enter the following settings:

- Password and Confirm Password: Enter hr for the password.
- Account Status: Select Unlocked.
- Roles: Ensure that both CONNECT and RESOURCE is enabled.

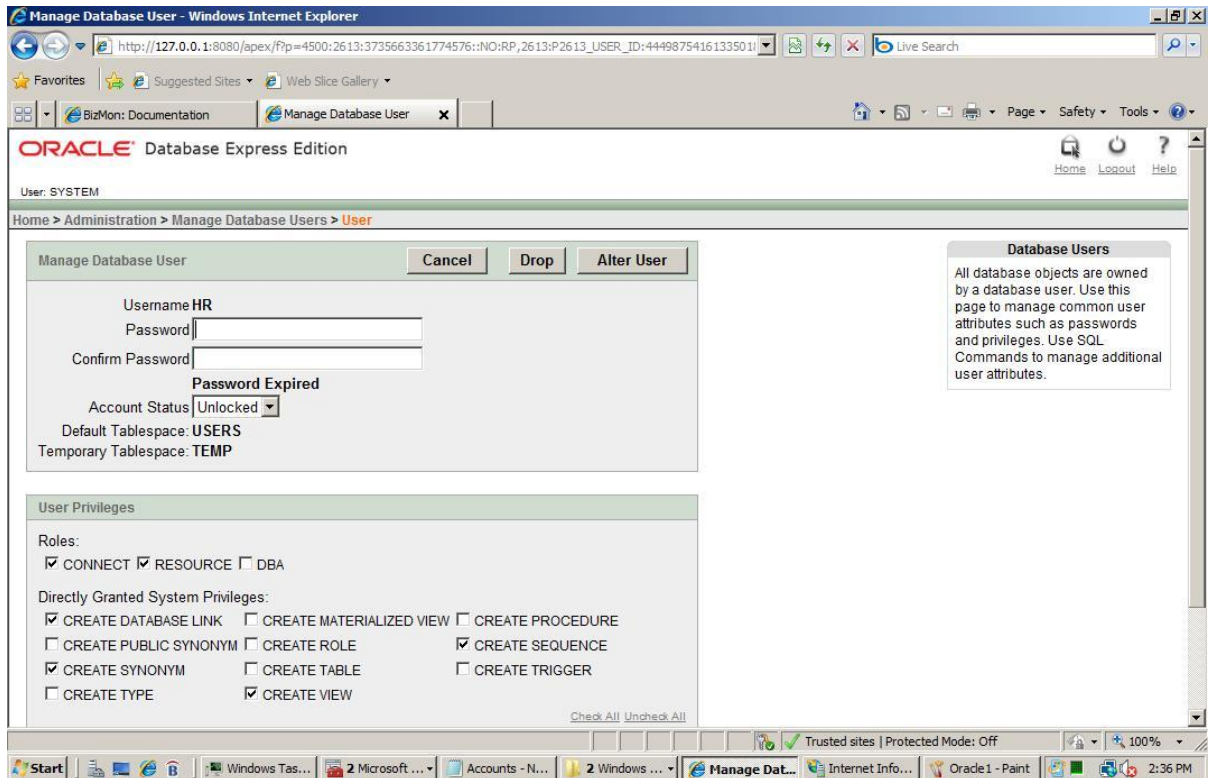


Figure 6: Screenshot of Oracle Database Express Edition: Manage Database User.

4. Click Alter User.

Once you have set this up you can check HR Schema by logging in with HR account and password. Go to Browser Objects (1) and select for instance Employees table (2) and then select data (3) and you will see what is shown below.

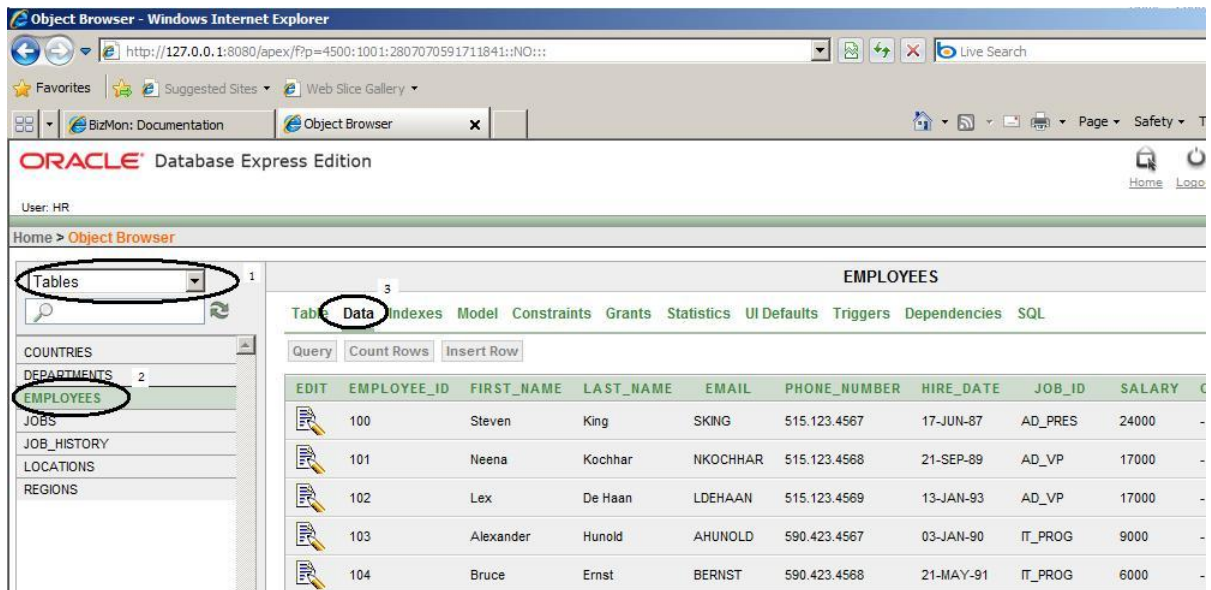


Figure 7: Screenshot of Object Browser showing data.

To work with the Adapter Pack 2.0 and Oracle you will need Oracle.DataAccess.dll version 2.111.7.0. This is included in Oracle 11g ODAC and Oracle Developer Tools for Visual Studio 11.1.0.7.20 and available through OTN. ODAC is a zip file you will need to extract. After extraction you are able to execute the setup. In setup you will need to select Oracle Data Access Components for Oracle Client 11.1.0.7.20. In next screen leave everything as is and proceed to available product components. You only have to select **Oracle Data Provider** component as shown below, and **Oracle Services For Microsoft Transaction Server** (additional Oracle Instant Client is selected by Wizard and is required!).

To start installation click next and install and check GAC afterwards to verify is correct Oracle.DataAccess.dll is present like picture below.

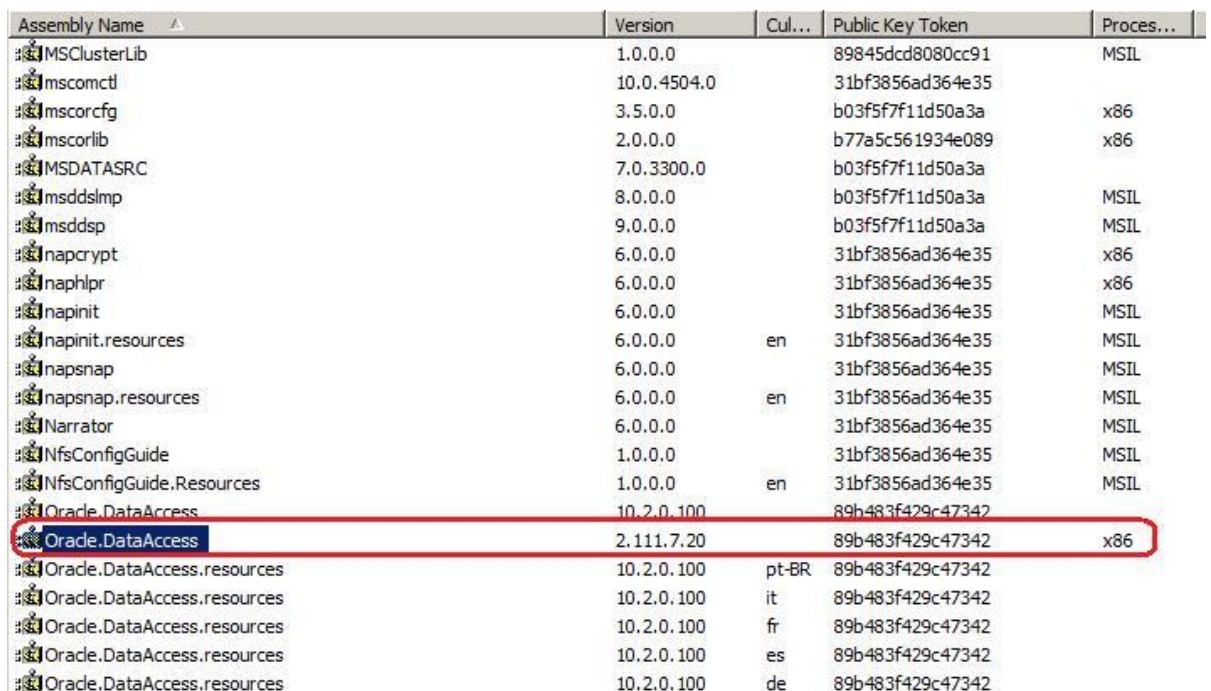


Figure 8: Screenshot of Assembly Cache showing Oracle.DataAccess assembly version.



Having done all the preparations above us we can now dive into how to build an application that retrieves data from backend system in this case Oracle. Through provided Oracle adapter interoperability can be achieved.



The BizTalk Adapter Pack 2.0 in action

Scenario and walkthroughs

We will walkthrough how to retrieve data from Oracle's test DEPARTMENTS table; firstly from a BizTalk perspective using an orchestration, schema and query message and then from a windows application perspective using a .NET client proxy code and a Windows form. Microsoft offers a wide variety of examples based on scenario's shown in picture 9.

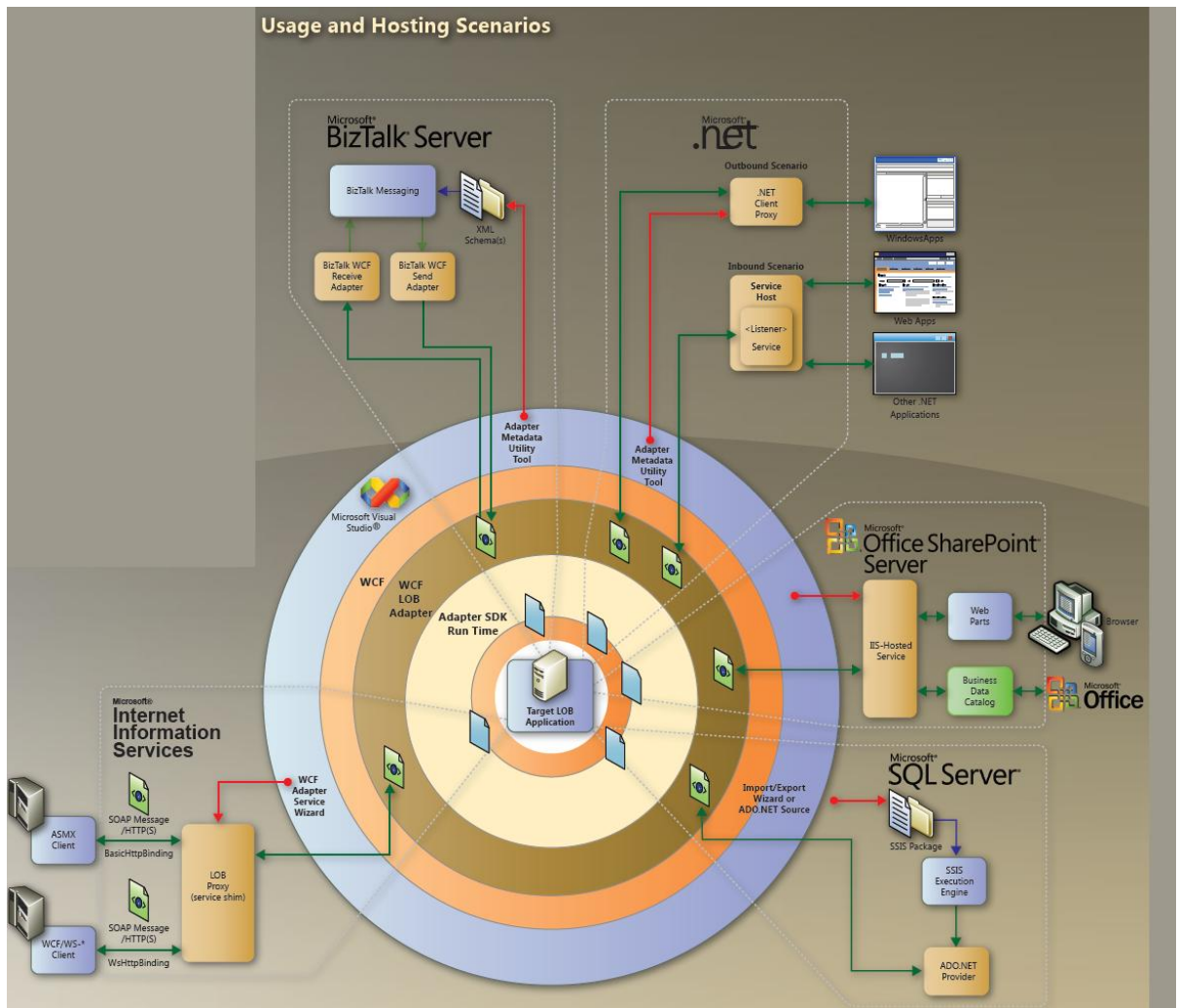


Figure 9: Usage and Hosting Scenario's.

Through this location: <http://msdn.microsoft.com/en-us/biztalk/dd796258.aspx>, the examples of scenario's depicted in picture 9 can be obtained. These samples are built with Visual Studio 2005 and need to be migrated if you want to apply them in a BizTalk Server 2009 environment. Walkthroughs described following paragraphs will provide for an example of integration with Oracle 10g Express Edition with:

1. Microsoft BizTalk Server using Metadata Utility tool to generate XML Schema's.
2. Microsoft .NET using Meta Utility tool to generate .NET Client Proxy.

Adapter with BizTalk Server Walkthrough

The following tasks need to be performed in Visual Studio:

1. Create a new Empty BizTalk project and Name it: HR_Messaging.
2. Go to your project and right-click, navigate to Add → Add Generated Items ... and click on it
3. Select Add Adapter Metadata, click Add
4. Select WCF-OracleDB from list of registered adapters and click Next
5. The Consume Adapter Service Window will pop up (**Adapter Metadata Utility Tool**) and you can select OracleDB binding and click on configure and pop up window will appear called configure adapter.

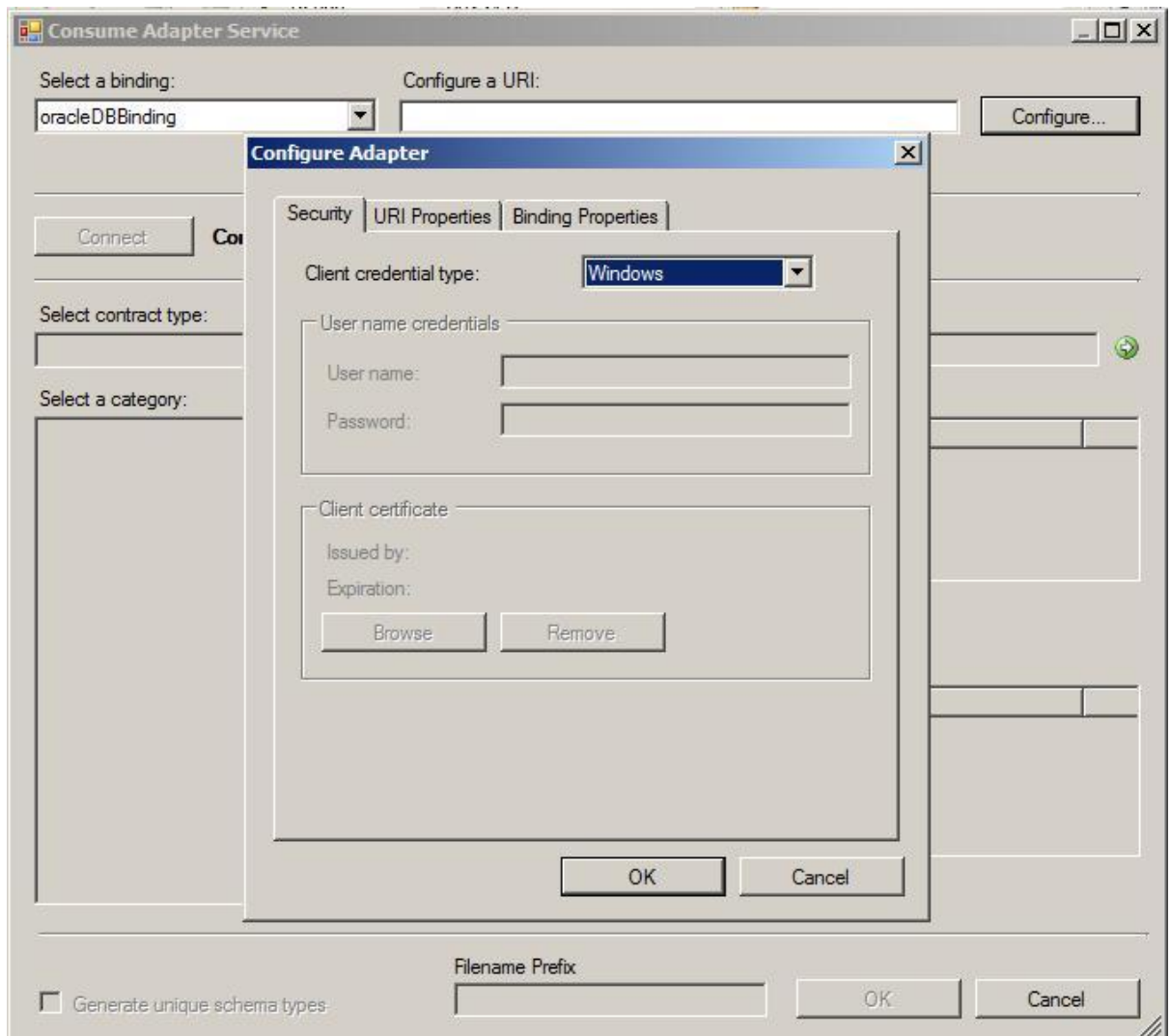


Figure 10: Screenshot of Configure Adapter Window.

6. Within Configure Window select on security tab **Username** for client credential type.
7. Fill in user name HR and password hr.

8. Select next Tab URI Properties and fill in the Server Address: that is your computer name and Service Name: XE and then click OK.
9. Click Connect and contract type and category should appear.
10. Select HR in category and click, and navigate to the DEPARTMENTS table.

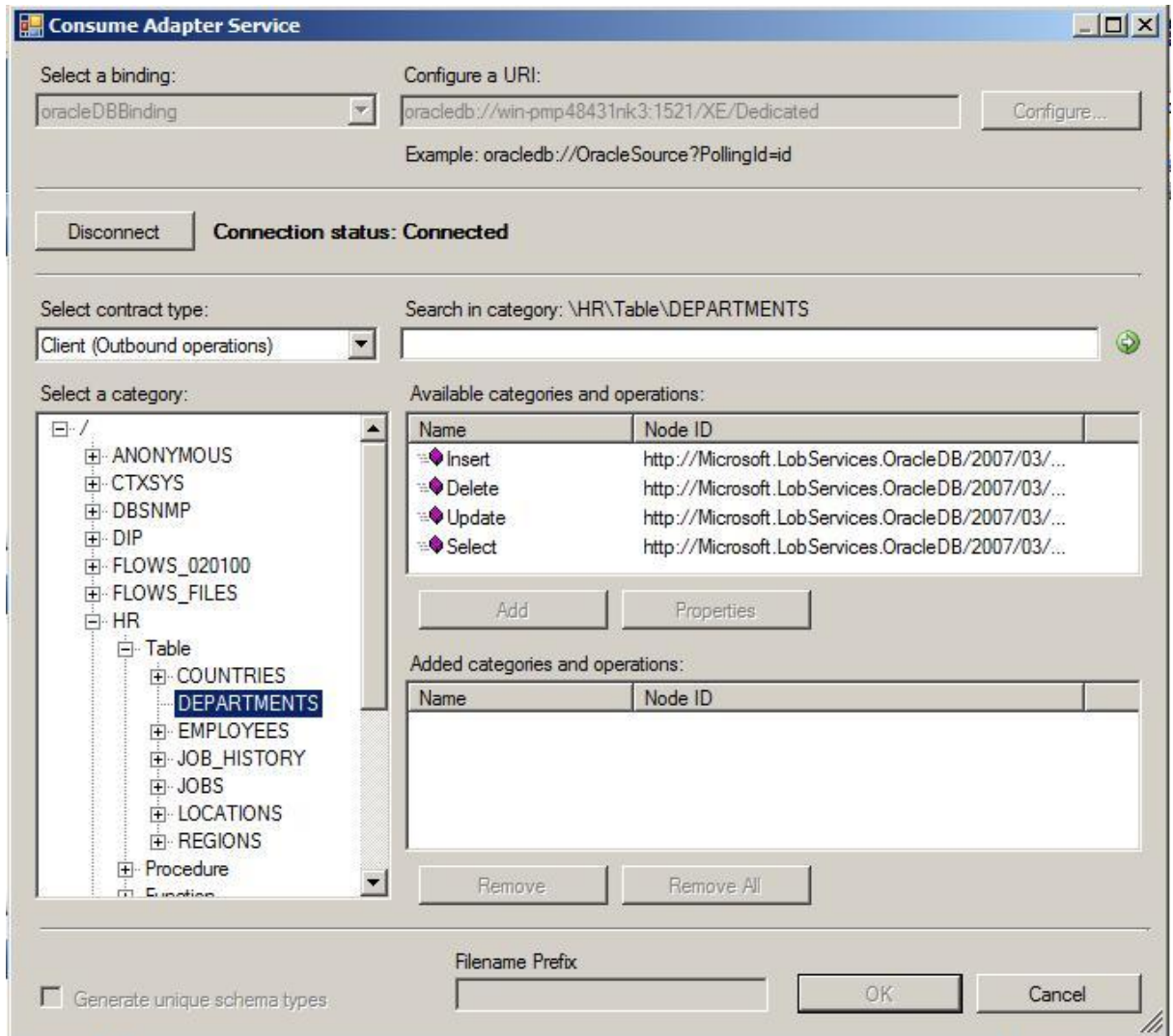


Figure 11: Screenshot of Consume Adapter Service Add In.

11. Choose Select in available categories and operations and click Add.
12. Click Ok and a schema will be generated.
13. Rename **BizTalk Orchestration** to **Department_Query.odx**.
14. Delete multi-message parts in orchestration viewer.
15. Place four shapes in the orchestration designer: Receive, Send, Receive, Send and name them accordingly:
 - a. ReceiveMessage
 - b. SendMessage



- c. ReceiveResponse
 - d. SendResponse
16. In the Orchestration View create two messages by right clicking on message and selecting add new message:
- a. First message is a request message:
 - i. Identifier: Request
 - ii. Message Type:
HR_Messaging.OracleDBBinding_DEPARTMENTS.Select
 - b. Second message is a response message:
 - i. Identifier: Response
 - ii. Message Type:
HR_Messaging.OracleDBBinding_DEPARTMENTS.SelectResponse
17. Next create three ports:
- a. Click new configured port:
 - i. Name: FileIn
 - ii. PortTypeName: FileInType
 - iii. Communication Pattern: One-Way
 - iv. Access Restriction: Internal – limited to this project
 - v. Port direction of communication: I'll always be receiving messages on this port
 - vi. Port Binding: Specify later
 - vii. Operation Identifier: Select
 - b. Click new configured port:
 - i. Name: LOBPort
 - ii. Use Existing Port: HR_Messaging.HR_Table_DEPARTMENTS
 - iii. Port direction of communication: I'll be a sending a request and receiving a response
 - iv. Port Binding: Specify later
 - c. Click new configured port:
 - i. Name: SaveResponse
 - ii. PortTypeName: SaveResponse Type
 - iii. Communication Pattern: One-Way
 - iv. Access Restriction: Internal – limited to this project
 - v. Port direction of communication: I'll always be sending messages on this port
 - vi. Port Binding: Specify later
 - vii. Operation Identifier: Select
18. Configure the shapes accordingly:



- a. ReceiveMessage Shape
 - i. Message: Request
 - ii. Activate: True
 - iii. Connect to FileIn Port (Request)
- b. SendMessage Shape
 - i. Message: Request
 - ii. Connect to LOBPort (Request)
- c. ReceiveResponse Shape
 - i. Message: Response
 - ii. Connect to LOBPort (Response)
- d. SendResponse Shape
 - i. Message: Response
 - ii. Connect to SaveResponse (Response)

19. Your orchestration should look like picture below.

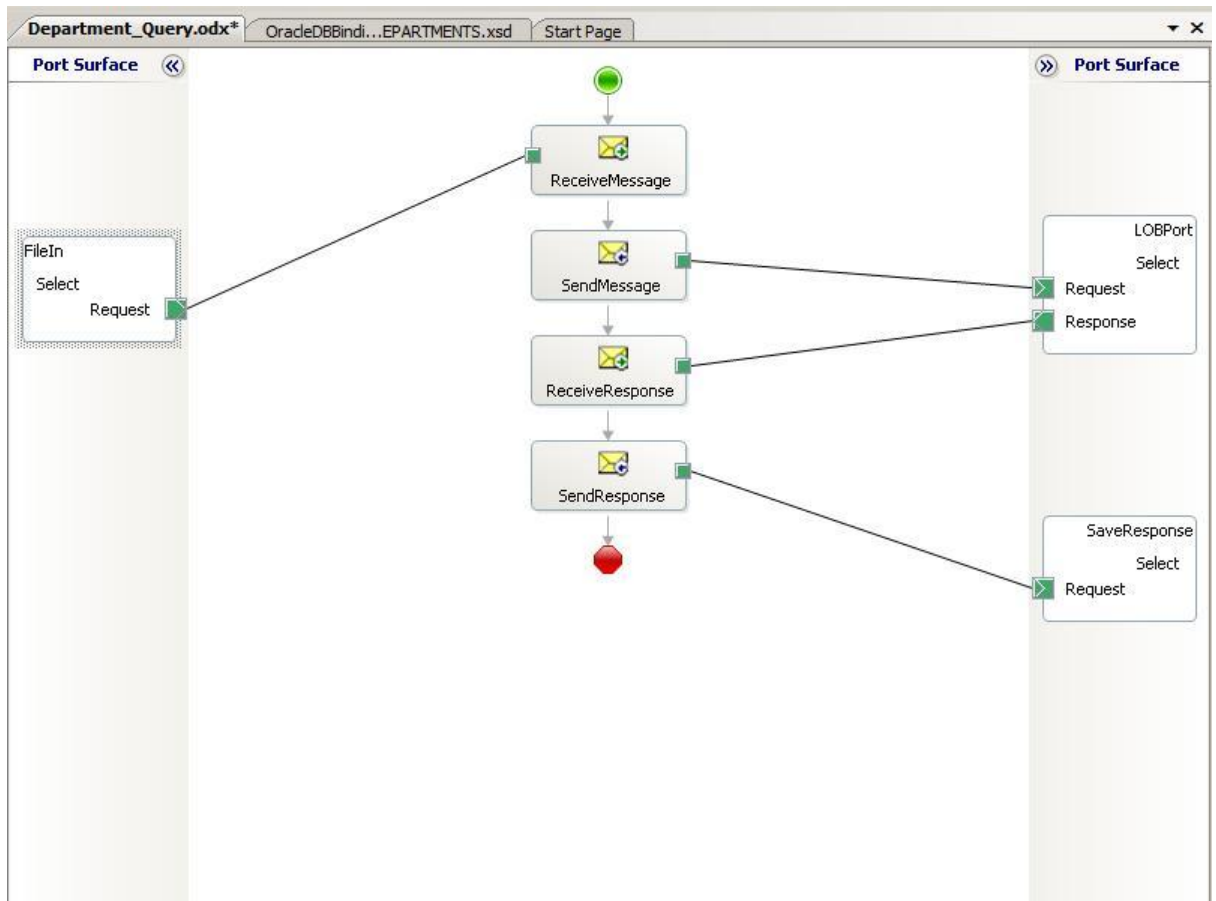


Figure 12: Screenshot of Department Query Orchestration.



20. Next build your solution.
21. Right-click your project in solution explorer and go to signing
22. Select sign assembly and click new
 - a. Name: HR_Messaging
 - b. Unselect Protect my key file with a password
 - c. Click Ok
23. Go to deployment tab and fill in Application Name: HR_Messaging
24. Right click on project and click Deploy.
25. Create an instance of Select Message by right clicking on your generated schema and click generate instance.

Message:

```
<ns0:Select  
xmlns:ns0="http://Microsoft.LobServices.OracleDB/2007/03/HR/Table/DEPARTMENTS">  
<ns0:COLUMN_NAMES>COLUMN_NAMES_0</ns0:COLUMN_NAMES>  
<ns0:FILTER>FILTER_0</ns0:FILTER>  
</ns0:Select>
```

26. Click on URL in Output Window and copy message to notepad
27. Save it as TestInstance.xml, with UTF-8 encoding in the folder where your solution resides.
28. Create in that folder two new folders called IN and RESPONSE.
29. Go to BizTalk Administration Console and open BizTalk Group.
30. Navigate to HR_Messaging Application and right-click to Configure.
31. Select Orchestration and select BizTalk Host.
32. Next configure ports:
 - a. New Receive Port:
 - i. Name: ReceivePort FileIn
 - ii. Receive Location: new
 1. Name: Receive Location FileIn
 2. Type: File
 - a. Configure file location (folder in solution directory called IN)
 3. Receive Pipeline: XML Receive
 - b. New Send Port
 - i. Name: Send Port LOB
 - ii. Type: WCF-Custom



33. Configure select on Binding Tab the Binding Type OracleDBBinding

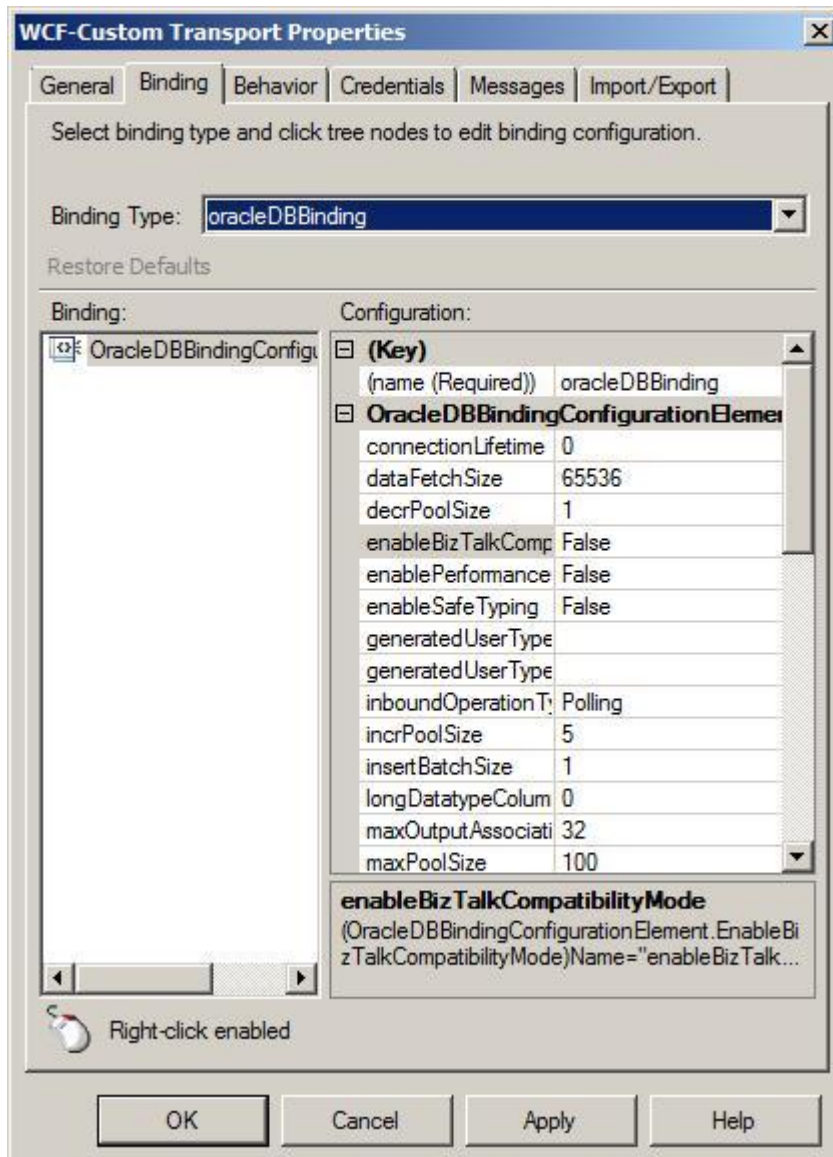


Figure 13: Screenshot of WCF-Custom Transport Properties Window.

- i. On General tab fill in URI: oracledb://<your machine name>
- ii. SOAP Action Header:

Action:

http://Microsoft.LobServices.OracleDB/2007/03/HR/Table/DEPARTMENTS/SELECT

- b. New Send Port:
 - i. Name: SendPort Response
 - ii. Type: File
 - 1. Configure file location (folder in solution directory called RESPONSE)

- iii. Receive Pipeline: PassThruTransmit
34. Finally right click application and click Start.
35. Check if proper access rights are applied to folders, if not do so by right click folders IN and RESPONSE and add BizTalk Application Users in Security Tab.
36. Modify Test Instance message by opening it and change as displayed below:

```
<Select  
xmlns="http://Microsoft.LobServices.OracleDB/2007/03/HR/Table/DEPA  
RTMENTS">  
  <COLUMN_NAMES>*</COLUMN_NAMES>  
  <FILTER>DEPARTMENT_NAME='IT'</FILTER>  
</Select>
```

37. Save the message as depicted above.
38. Copy TestInstance.Xml to IN folder
39. Browse to the RESPONSE folder and a message similar to the one below should have been received as a response:



Figure 14: Screenshot of response message.

Adapter with .NET Walkthrough

The following tasks need to be performed in Visual Studio:

1. Create a new Windows Forms Application project and Name it: HRApplication.
2. Navigate to Form1 in your project and rename it to DEPARTMENT SEARCH.
3. In property windows of form rename Text to Department Search
4. On form place the following controls:
 - a. Label
 - i. Name: lblDepartment
 - ii. Text: Department
 - b. Textbox
 - i. Name: txtDepartment
 - c. Button
 - i. Name: btnSearch
 - ii. Text: Search
 - d. Label
 - i. Name: lblResult
 - ii. Text: Result
 - e. Textbox
 - i. Name:txtResult
 - ii.
5. Form should look like something below

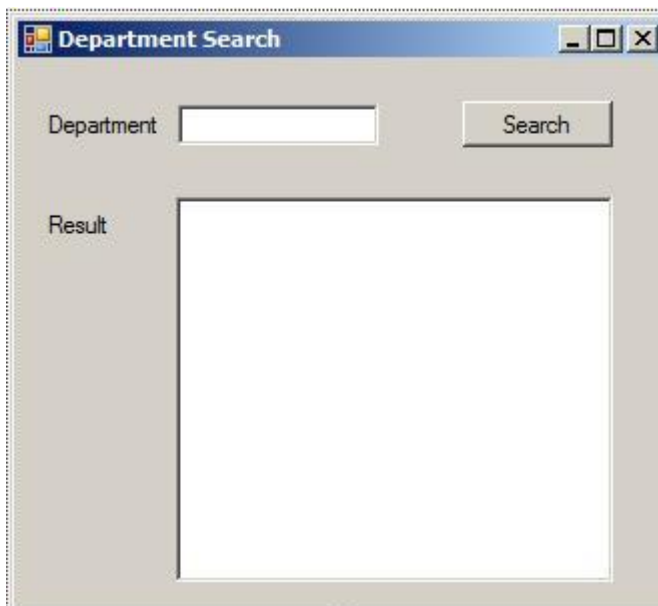


Figure 15: Screenshot of Windows form.





6. Next right click you project and click Add Adapter Service Reference ...
7. The Add Adapter Reference Window will pop up (Adapter Metadata Utility Tool) and you can select OracleDB binding and click on configure and pop up window will appear called configure adapter.
8. Within Configure Window select on security tab **Username** for client credential type.
9. Fill in user name HR and password hr.
10. Select next Tab URI Properties and fill in Server Address: that is your computer name and Service Name: XE and then click OK.
11. Click Connect and contract type and category should appear.
12. Select HR in category and click, and navigate to tables to DEPARTMENTS.
13. Choose Select in available categories and operations and click Add.
14. Click Ok and a .NET class OracleDBBindingClient.cs will be generated.
15. The generated class acts as a .NET Client Proxy. Now go to Windows Form and click on Search button.
16. Paste the following code in the code behind:

using microsoft.lobservices.oracledb._2007._03.HR.Table.DEPARTMENTS;

17. Paste the following code inside the brackets of btn_Search_Click():

```
private void btnSearch_Click(object sender, EventArgs e)
{
    DEPARTMENTSRECORDSELECT[] selectRecords = new DEPARTMENTSRECORDSELECT[0];

    // Initialize WCF client variable
    HR_Table_DEPARTMENTSClient department=null;

    try
    {
        // initialize the DEPARTMENTS table client
        department = new HR_Table_DEPARTMENTSClient("OracleDBBinding_HR_Table_DEPARTMENTS");
        //Security Credentials
        department.ClientCredentials.UserName.UserName = "HR";
        department.ClientCredentials.UserName.Password = "hr";
        //Open Client
        department.Open();
        // Select all records and write them to the textbox result
        selectRecords = department.Select("*", "DEPARTMENT_NAME = '" + txtDepartment.Text + "'");
        txtResult.Text = "Department_ID : " + selectRecords[0].DEPARTMENT_ID + "\r\n" + "Location_ID : " +
            selectRecords[0].LOCATION_ID + "\r\n" + "Manager_ID : " + selectRecords[0].MANAGER_ID;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        department.Close();
    }
}
```

18. Build solution by right clicking project in solution explorer and click build.
19. Click F5 to run.
20. Fill in **IT** in Department and Click Search.
21. Result should look like screen below:



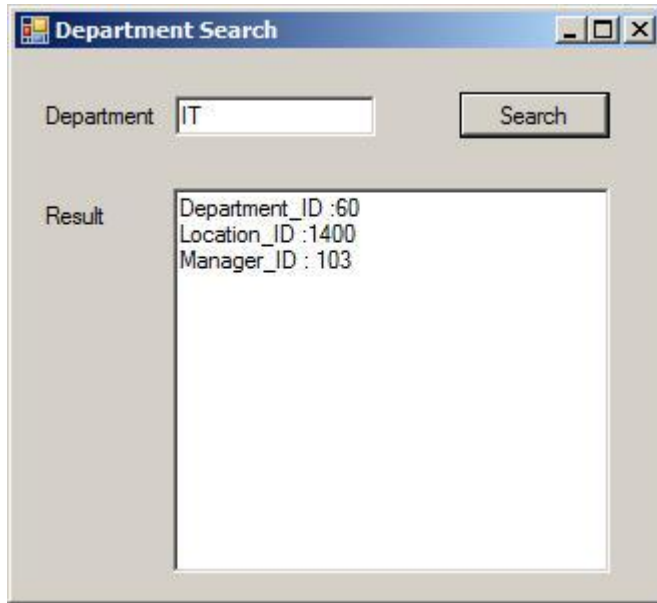


Figure 16: Screenshot of Windows form performing a search.





Conclusion

In this article I gave some historical background on BizTalk's LOB Adapters while focusing on the current BizTalk Adapter Pack 2.0 and its WCF programming model. Two walkthroughs provided Adapter Pack usage scenarios; one using BizTalk Server and the other using Microsoft .NET. The Adapter Pack provides versatile and consistent interoperability with LOB systems like SAP, Siebel, Oracle, and TIBCO. Data can be exposed through adapters to BizTalk Server, .NET, MOSS, SQL Server or IIS depending on usage/hosting scenarios. Migration of solutions between these different usage scenarios is made easier due to the shared use of the WCF programming model. Do bear in mind that the version of the LOB system you want to integrate with depends on supported versions of Adapter Pack 2.0. If you require interoperability with older versions of a LOB system you may have to turn to a non-WCF LOB Adapter as mentioned in this article.

Resources

Microsoft Adapter Pack 1.0:

<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=58bd17c8-8efa-4b69-ab51-9f5b11a962dc>

Microsoft BizTalk Adapter Pack 2.0 Installation Guide:

<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=ff0b19a6-72c1-43b3-9d38-b874303590ce>

BizTalk Adapter Pack 2.0 now Released!!!:

<http://blogs.breezetraining.com.au/mickb/2009/05/06/BizTalkAdapterPack20NowReleased.aspx>

WCF LOB Adapter SDK:

<http://www.microsoft.com/downloads/details.aspx?familyid=56278FDE-B708-469C-987E-DED9C6C5E580&displaylang=en>

BizTalk Adapter Pack 2.0: <http://www.microsoft.com/biztalk/en/us/adapter-pack.aspx>

Oracle 10g Express Edition:

<http://www.oracle.com/technology/products/database/xe/index.html>

BizTalk Adapter Pack 2.0 Oracle Binding: <http://soa-thoughts.blogspot.com/2009/08/biztalk-adapter-pack-20-oraclebinding.html>

Microsoft Adds Eight Application Adapters for BizTalk Server:

<http://www.microsoft.com/presspass/press/2005/aug05/08-02applicationadapterspr.msp>

WCF LOB Adapter SDK and BizTalk Adapter Pack:

<http://blogs.msdn.com/adapters/default.aspx>

Whitepaper: WCF LOB Adapter SDK and the BizTalk Adapter Pack:

<http://www.pluralsight.com/community/blogs/aaron/archive/2009/04/27/whitepaper-on-wcf-lob-adapter-sdk-and-the-biztalk-adapter-pack.aspx>





Acknowledgements

I would like to thank my colleagues Sander Nefs, Gary Daly and Tomasso Groenendijk from motion10 helping through auditing and editing this article. Thanks a lot guys!

About motion10

motion10 is a global provider of consultancy services geared to complex integrated environments. Its service offering also includes implementation, support and training. Unlike other systems integrators motion10 focuses exclusively on the Microsoft Application Platform which supports products such as BizTalk Server, SharePoint Server and SQL Server. The company has its own experts who also advise Microsoft in the further development of these products. Headquartered in Rotterdam (the Netherlands), motion10 has offices in Atlanta and Seattle, and an extensive partner network across Europe, North America and Asia. The motion10 consultants have completed a total of more than 500 BizTalk Server implementations. For more information: www.motion10.com.

