How To Access SAP Business Data From Silverlight 4 Clients Using WCF RIA Services

Jürgen Bäurle

November 2010

Parago Media GmbH & Co. KG

Introduction

The introduction of Microsoft's WCF RIA Services for Silverlight 4 simplified very much the development process of N-tier business applications using Silverlight and ASP.NET. By using this new technology we can also easy access and integrate SAP business data in Silverlight clients.

This article shows how to provide a SAP domain service as web service that will be consumed by a Silverlight client. The sample application will allow the user to query customer data. The service uses LINQ to SAP from Theobald Software to connect to a SAP R/3 system.

Project Setup

The first step in setting up a new Silverlight 4 project with WCF RIA Services is to create a solution using the Visual Studio template *Silverlight Navigation Application*:



Visual Studio 2010 is then asking you to create an additional web application, which hosts the Silverlight application. It's important to select the checkbox *Enable WCF RLA Services* (see screenshot below):

New Silverlight Application
Click the checkbox below to host this Silverlight application in a Web site. Otherwise, a test page will be generated during build.
Host the Silverlight application in a new Web site
New Web project name:
SAP2Silverlight.Web
New Web project type:
ASP.NET Web Application Project
Options
Silverlight Version:
Silverlight 4 🗸
Enable WCF RIA Services
OK Cancel

After clicking the Ok button, Visual Studio generates a solution with two projects, one Silverlight 4 project and one ASP.NET project. In the next section we are creating the SAP data access layer using the LINQ to SAP designer.

LINQ to SAP

The LINQ to SAP provider and its Visual Studio 2010 designer offers a very handy way to design SAP interfaces visually. The designer will generate the code for the SAP data access layer automatically, similar to LINQ to SQL. The LINQ provider is part of the .NET library ERPConnect.net from Theobald Software. The company offers a demo version for download on its homepage.

The next step is to create the needed LINQ to SAP file by opening the Add New Item dialog:

Installed Templates	Sort by: Defaul	•] 💷		Search Installed Templates	\$
▲ Visual C#	্ৰ	Silverlight Application		LINQ to ERP	Type: Visual C#	
Data		Silverlight-enabled WCF	ervice		Adds a LINQ to ERP file to	the project.
General Web	<u></u>	Site Map				
Windows Forms		Skin File		· \		
Extensibility		SQL Server Database				
Reporting		Text File				
Silverlight tangible modeling tools		Text Template				
tangible T4 Editor		WCF Data Service				
XNA Game Studio 4.0	3	WCF Service				
Online Templates		Web Configuration File				
	1	Web Service				
	2	Windows Script Host				
	(11)	XML File				
	品	XML Schema				
	لع	XSLT File				
	B	Activity				
	-	WCF Workflow Service				
	4				•	
Name: LINQtoERP1.e	erp					
					Add	Cancel

LINQ to SAP is internally called LINQ to ERP.

Clicking the Add button will create a new ERP file and opens the LINQ designer. Now, drag the Function object from the toolbox and drop it onto the designer surface. If you have not entered the SAP connection data so far, you are now asked to do so:

👓 SAP2Silverlight - Microsoft Visual	Studio (Administrator)	
File Edit View Telerik Project	Build Debug Team Data Tools VisualSVN Test Analyze Window Help	
i 🔂 • 🛅 • 😂 🖌 🕼 🗸 🐁 🛛	🚵 🧐 🔹 🖓 📲 📮 🖳 💺 🕨 Windows Phone 7 Emulator 🕞 🖬 Debug 🔹 🛛 Any CPU 🔹 🦉 #region	i enums 🕞 🚽 🖓 🕾 📷 💥 💥 🛃 🚆
Toolbox • # X		Solution Evolorer 🚽 🗸 🕹
✓ LINO to ERP	Lingtockritelp ×	
Pointer		Selection (CAD2City reliants)
III Table		Solution SAP2Silverlight
Y Ouerv	Connection	Properties
Function		References
1 IDoc	LING to EKP Connection	Assets
BWCube	[LINQ TO ERP Info]	Views
▲ General		App.xaml
1	Client 800	A SAP2Silverlight.Web
There are no usable controls in	Licensen TECTICED	Properties
this text to add it to the toolbox.		References
	Password	ClientBin
	Language EN	LINQtoERP1.erp
	Extensions	LINQtoERP1.erp.diagram
		SAP2SilverlightTestPage.aspx
	Use Single Application Server O Use Load Balancing	UNOt Solutio
	Host TESTSYSTEM MessageServer	
	SystemNumber 01 Group / SID	
	Use HTTP / Web Service	
	HTTP Url	Class SAPContext
		ConnectionString
	Test Connection Cancel Ok	Create Objects Outside Of Ci True
		Modifier Public
		Namespace
		Reuse Classes False *
		ConnectionString
🚆 Server Explorer 🔆 Toolbox	· · · · · · · · · · · · · · · · · · ·	
📸 Error List 🧮 Output 🔉 Find Sy	/mbol Results	
Ready		

Enter the connection data for your SAP R/3 system and then click the Ok button. Next, search for and select the SAP function module named *SD_RFC_CUSTOMER_GET*. The function module provides a list of customer data.

The RFC Function modules dialog opens and let you define the necessary parameters:

RFC Functio	n Module	es / BAPIs					
8 8	F	RFC Function Modules /	BAPIs				
Function Method	SD_F GetC	RFC_CUSTOMER_GET	Ø	Modifier	Public	•	
Exports	Imports	Changing Tables				1	
Pass	Туре	Name	Variable	Va	alue	Structure Name	Edit
	String						Edit
1			1				
				Cancel		Ok	

In the above function dialog, change the method name to *GetCustomers* and mark the *Pass* checkbox for the *NAME1* parameter in the *Exports* tab. Also set the variable name to *namePattern*. On the *Tables* tab mark the *Return* checkbox for the table parameter *CUSTOMER_T* and set the table and structure name to *CustomerTable* and *CustomerRow*.

RFC Function	Modules / BAPIs			_	
8	RFC Function Modules [LINO To ERP Info]	/ BAPIs			
Function Method	SD_RFC_CUSTOMER_GET GetCustomers	Ø	Modifier Public	•	
Exports I	mports Changing Tables	Variable	Table Name	Shushing Name	E dia
Return		variable	CustomerTable	CustomerRow	Edit
1			1		
			Cancel	Ok	

After clicking the Ok button and saving the ERP file, the LINQ designer will generate a *SAPContext* class which contains a method called *GetCustomers* with an input parameter named *namePattern*. This method executes a search for SAP customer data allowing the user to enter a wildcard pattern. The method returns a table of customer data:



On the LINQ designer level (click on the free part of the LINQ designer surface) the property Create Object Outside Of Context Class must be set to True:

Co SAP2Silverlight - Microsoft Visual	Studio (Administrator)	
File Edit View Telerik Project	Build Debug Team Data Tools VisualSVN Test Analyze Window Help	
🛅 • 🛅 • 💕 📕 🥥 🐰 🔖	🖹 🧐 🔹 🖓 📲 📮 🖳 🕨 Windows Phone 7 Emulator 🚽 💷 Debug 🔹 Any CPU 🔹 🧭 #region	enums 🔹 🔹 🖓 🕾 🖬 😫 🖡
Toolbox - I ×	LINQtoERP1.Designer.cs	Solution Explorer 👻 🖣 🗙
LINQ to ERP		💁 🖫 🗿 🔊 👘
Pointer		Solution 'SAP2Silverlight' (2 projects)
III Table		a 🥶 SAP2Silverlight
Query		Properties
Function		Accets
1 IDoc		Views
BWCube	Function:	App.xaml
a General	SD RFC CUSTOMER GET	MainPage.xaml
There are no usable controls in		A Carling SAP2Silverlight.Web
this group. Drag an item onto		References
this text to add it to the toolbox.		🗀 ClientBin
		▲ LINQtoERP1.erp
		LINQtoERP1.Designer.cs
		SAP2SilverlightTestPage.aspx
		Capital Capital Character Control
		Properties • 4 ×
		LINQtoERP1 ERPModel
		Class SAPContext ^
		Connectionstring USER=TESTUSER P
		Create Parameterless Constri True
		Modifier Public
		Namespace
		Reuse Classes False *
		Class
💐 Server Explorer 🔀 Toolbox	۰ ۳	
🚼 Error List 🥫 Output 🛤 Find S	/mbol Results	
Ready	Ln1	Col 1 Ch 1 INS!

Now, we finally add a *Customer* class which we use in our SAP domain service later on. This class and its values will be transmitted to the Silverlight client by the WCF RIA Services. It's important to set the Key attribute on the identifier fields for WCF RIA Services, otherwise the project will not compile:



That's it! We now have our SAP data access layer ready to use and can start adding the domain service in the next section.

SAP Domain Service

The next step is to add the SAP domain service to our web project. A domain service is a specialized WCF service and is one of the core constructs of WCF RIA Services. The service exposes operations that can be called from the client generated code. On the client side we use the domain context to access the domain service on the server side.

nstalled Templates	Sort by: Default	- 💷 💷	Search Installed Templates
Visual C#	Web Form	Browser File	Type: Visual C#
Data	Web Form using Master Page	Class Diagram	Domain service class for WCF RIA Service
General Web	Web User Control	Code Analysis Rule Set	applications
Windows Forms	Class	🕙 Code File	
WPF Extensibility	Master Page	🕎 Crystal Report	
Reporting	Nested Master Page	👸 DataSet	
Silverlight tangible modeling tools	HTML Page	🤯 Debugger Visualizer	
tangible T4 Editor	Style Sheet	Domain Service Class	
Workflow XNA Game Studio 4.0	JScript File	🚱 Dynamic Data Field	
Inline Templates	Interface	🖹 Generic Handler	
	ADO.NET Entity Data Model	Global Application Class	
	ADO.NET EntityObject Generator	Installer Class	
	ADO.NET Self-Tracking Entity	LINQ to SQL Classes	
	AJAX-enabled WCF Service	Preprocessed Text Template	
	Application Manifest File	🔝 Report	
	Assembly Information File	🔊 Report Wizard	
	Authentication Domain Service	Resources File	
	۰ III		4
lame: SAPService.c	s		

Add a new Domain Service Class and name it SAPService:

In the upcoming dialog create an empty domain service class by just clicking the Ok button:

Add New Domain Service Class	8
Domain service class name:	
SAPService	
Enable client access	
Expose OData endpoint	
Available DataContext/ObjectContext classes:	
<empty class="" domain="" service=""></empty>	•
Entities	Enable editing
Generate associated classes for metadata	OK Cancel

Next, we add the service operation *GetCustomers* to the SAP service with a name pattern parameter. The operation then returns a list of *Customer* objects. The Query attribute limits the result set to 200 entries.

The operation uses the visually designed SAP data access logic to retrieve the SAP customer data. First of all, an instance of the *SAPContext* class will be created using a connection string (see sample in code). For more details regarding the SAP connection string see the ERPConnect.net manual.

The LINQ to SAP context class contains the *GetCustomers* method which we will call using the given namePattern parameter. Next, the operation creates an instance of the *Customer* class for each customer record returned by SAP.

The license code for the ERPConnect.net library is set in the constructor of our domain service class.



That's all we need on the server side.

In the next section we are implementing the Silverlight client.

Silverlight Client

The implementation of the client side is straightforward. The home view contains a DataGrid control to display the list of customer data as well as a search area with TextBox and Button controls to allow users to enter name search pattern.

The click event handler of the load button, called *OnLoadButtonClick*, will execute the SAP service. The boilerplate code to access the web service was generated by WCF RIA Services in the subfolder *Generated_Code* in the Silverlight project.

First of all, an instance of the *SAPContext* will be created. Then we load the query *GetCustomersQuery* and execute the service operation on the server side using WCF RIA Services. If the domain service returns an error, the callback anonymous method will mark the error as handled and display the error message.

If the execution of the service operation succeeded the result set gets displayed in the DataGrid control.

Home.xaml.c	s 🗙 Homexaml 👻
😤 SP2010SA	PToSilverlight.Home - 📓 OnLoadButtonClick(object sender, RoutedEventArgs e) -
9	{
10 🗉	public Home()
11	{
12	InitializeComponent();
13	}
14	
15	void OnLoadButtonClick(object sender, RoutedEventArgs e)
16	
1/	LoadButton.istnabled = taise;
18	stming normalistons - stming TelullOelbitsEnges(NempOsttonsToutPoy Tout)) "#" - NempOsttonsToutF
20	String namerattern = String.Isutionwhitespace(wameratterniextbox.iext) ? * : wameratterniex =
20	SAPContext context = new SAPContext().
21	Shi concext concext - new Shi concext();
23	context.load(context.GetCustomersQuery(namePattern), delegate(loadOperation <customer> operation</customer>
24	
25	if(operation.HasError)
26	
27	operation.MarkErrorAsHandled();
28	MessageBox.Show(operation.Error.Message);
29	}
30	else
31	CustomerList.ItemsSource = operation.Entities;
32	
33	LoadButton.IsEnabled = true;
34	
35	<pre>}, null);</pre>
36	
100 % 👻	

The next screenshot shows the final result:

🏉 Home - Windo	ws Internet Explorer	- a	- 1 mg (24)			1 23
@ ~ 	http://localhost:50002/Default.aspx#/Home		🕶 🗟 🐓 🗙 🔁	Bing		- م
🚖 Favoriten	🏉 Home		🐴 🕶 🗟 👻 🖻	🛛 🖶 🔻 Seite 🕶	Sicherheit 👻 Extras	• 🕐 • »
👌 SAP	Customer Data				home abou	ıt
Hon Enter Custo	the name of the SAP customer or use mer Name: T*	the wildcard sign (*) t	to search for name pattern.			
ID	Name					
0000	003279 Tech Store					
0000	003287 Tornado Motorsports					
0000	007004 TetPak Inc.					
0000	007021 TelcoShop4U					
0000	007600 Telecomunicaciones Star S.A.					
0000	001600 TALPA GmbH					
0000	003380 Technical Expert Manufacturing					
0000	003390 Toro Motor Company					
0000	001390 Technik und Systeme GmbH					
0000	001780 The Bikes Shop					
0000	003474 TCC AG					
	DODIER TECHNA				•	
<u> </u>						
Fertig		👊 Lokales	Intranet Geschützter Modus: In	aktiv	🖓 👻 🔍 105	i % ▼

That's it.

Summary

This article has shown how easy SAP customer data can be integrate within Silverlight clients using tools like WCF RIA Services and LINQ to SAP. It is quite simple to extend the SAP service to integrate all kind of operations.

Contact Information

If you have any feedback or suggestions, please feel free to contact me:

Jürgen Bäurle jbaurle@parago.de http://www.parago.de/jbaurle

Parago Media GmbH & Co. KG Im Wengert 3 | 71336 Waiblingen, Germany | Phone +49.7146.861803 | Internet <u>http://www.parago.de</u>