[Typed Polling with WCF Adapters | Tallan](https://www.tallan.com/blog/2014/12/23/typed-polling-with-wcf-adapters/)

Typed Polling with WCF Adapters

Posted on December 23, 2014

[1 Comment](https://www.tallan.com/blog/2014/12/23/typed-polling-with-wcf-adapters/#comments)

Cesar Cordero

Typed Polling with WCF Adapters

It is very common to have a requirement in middleware design where we have to query [SQL](https://www.tallan.com/sql-server-2016/) tables and retrieve some data when certein conditions are met.

The [WCF](http://social.technet.microsoft.com/wiki/contents/articles/3956.wcf.aspx)-based adapters available with the BizTalk Adapter Pack enable the adapter clients to receive data-change messages from the SQL Server database(s). The adapter(s) support receiving “polling-based” messages wherein the adapter executes a specified SQL statement (SELECT statement or stored procedure), retrieves or updates the data, and provides the result to the adapter client at regular intervals of time. The adapters supports three types of polling:

* Weakly-typed Polling (also called **Polling**),
* Strongly-typed Polling (also called **TypedPolling**)
* Xml-Polling polling using statements or procedures that include a FOR XML clause

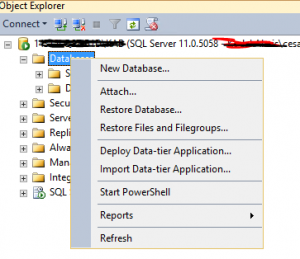
For more information about the different types of polling, see [Polling SQL Server by Using the SQL Adapter with BizTalk Server](http://go.microsoft.com/fwlink/?LinkId=221195)(http://go.microsoft.com/fwlink/?LinkId=221195).

Assumptions and prerequisites

This document is not intended as an MS [SQL](https://www.tallan.com/sql-server-2016/) Primer and assumes that the reader has basic Database administrative skills to create and manage databases, tables and stored procedures. It is also assumed that the reader has basic knowledge of BizTalk, BizTalk administration and basic deployment methodologies around the same.

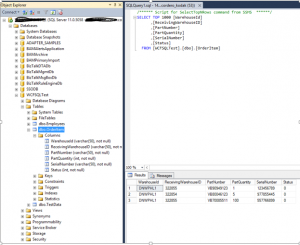
Create a new Database called **WCFSQLTest**

For the purposes of our discussion we will be creating a test database. Using SQL Server Management studio right click on the Database object and select new:

[](http://blog.tallan.com/wp-content/uploads/2014/12/CreateWCFSQLTest.png)

Create a new table in the **WCFSQLTest** DB called OrderItem:

Using SQL Server Management Studio and from within the newly created Database object:

[](http://blog.tallan.com/wp-content/uploads/2014/12/CreateNewTableOrderItem.png)

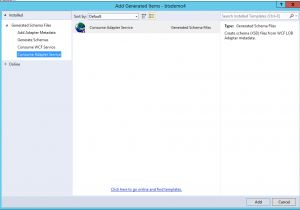
Launch Visual Studio

Launch visual studio and create a new empty BizTalk project called BTSDemo4

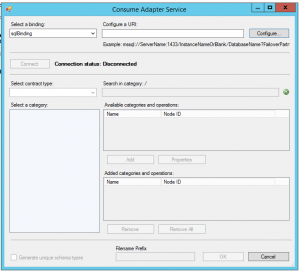
Add Generated Items:

From a newly  created BTSDemo4 solution/project right click on the project and choose “Add Generated Items” and choose

Consume Adapter Service from the Add Generated Items Wizard.

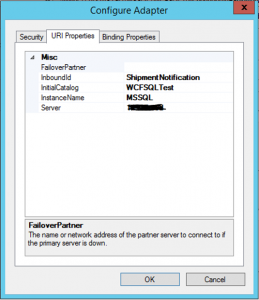
[](http://blog.tallan.com/wp-content/uploads/2014/12/ConsumeAdapter.png)

After clicking ADD then choose [SQL](https://www.tallan.com/sql-server-2016/)Binding from the “select a binding” drop down and then press the configure button:

[](http://blog.tallan.com/wp-content/uploads/2014/12/ConsumeAdapterWizard.png)

Configure URI:

Choose the URI Properties and fill them out according to your DBName (initial catalog), [SQL](https://www.tallan.com/sql-server-2016/) Instance name and server name. on the Inbound ID use a unique identifier related to the interface:

[](http://blog.tallan.com/wp-content/uploads/2014/12/ConfigureURI.png)

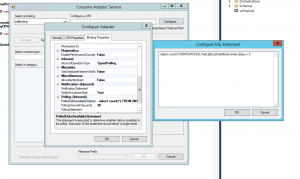
Binding Properties:

PollingDataAvailableStatement

After filling out the the URI PROPERTIES SECTION select the binding properties TAB and cut and paste the following select statement in the “PollingDataAvailableStatement” to determine if any records are available in our table:

select count(\*) FROM [WCF[SQL](https://www.tallan.com/sql-server-2016/)Test].[dbo].[OrderItem] where status = 0 where status =0 is NOT INTERFACED OR Processed by BizTalk.

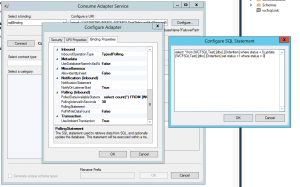
**Note**: PLEASE BE SURE the Inbound OperationType is set to TypedPolling

[](http://blog.tallan.com/wp-content/uploads/2014/12/BindingProperties1.png)

Polling Statement

Place the following [SQL](https://www.tallan.com/sql-server-2016/) in the “Polling Statement” section:

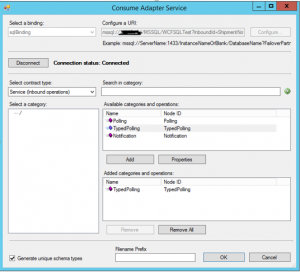
select \* from [WCF[SQL](https://www.tallan.com/sql-server-2016/)Test].[dbo].[OrderItem] where status = 0;update [WCFSQLTest].[dbo].[OrderItem] set status =1 where status = 0

[](http://blog.tallan.com/wp-content/uploads/2014/12/PollingStatement.png)

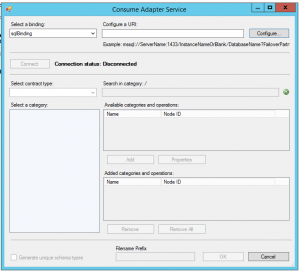
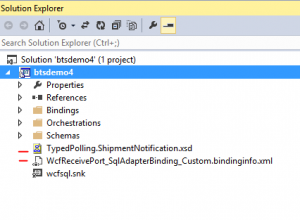
Then click OK

Connecting to the Database

Click connect to ensure the URI configuration is correct then be sure and select SERVICE INBOUND in the dropdown for the contract type on left side of screen. Then Choose TypedPolling and click add. Also choose Generate unique Schema type. Then click OK

[](http://blog.tallan.com/wp-content/uploads/2014/12/ConnectingToDatabase.png)

This should add two files to your solution:

[](http://blog.tallan.com/wp-content/uploads/2014/12/ConsumeAdapterWizard.png) [](http://blog.tallan.com/wp-content/uploads/2014/12/AddTwoFiles1.png)

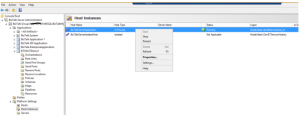
Please build the solution to be sure nothing has “broken”.

Deploy the solution from visual studio to your local BTS instance.

Import the WcfReceivePort\_SqlAdapterBinding\_Custom.bindinginfo.xml binding file into the currently deployed WCF Test solution on your local BizTalk Instance

Restarting Host Instances:

Under platform settings from the BizTalk admin console restart the local host instances:

[](http://blog.tallan.com/wp-content/uploads/2014/12/RestartingHostInstance.png)

Turn on BizTalk Application

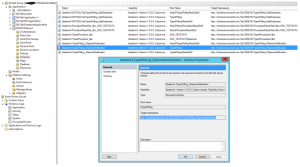
Turn on the receive location and then review the event log for errors.

We should find one subscription not found error that illustrated the pub/sub nature of BizTalk interfaces.

We will need to place a filter on a newly created send port to “subscribe” to the message type received in by our new receive location that polls the SQL table based on the schema generated by the Wizard in previous step and now deployed into our application via Visual Studio.

**BizTalk Schemas Target Namespaces:**

If we review the “schemas” section under our application tree in the BizTalk Admin console we will fine our schema there with a root name of “TypedPolling”:

[](http://blog.tallan.com/wp-content/uploads/2014/12/BizTalkSchemasTargetNameSpace.png)

**Subscription Filters**:

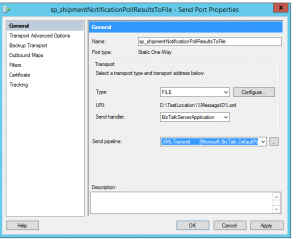
The target name space and the root name are the two key pieces to creating a filter in a send port that will act as the subscribing artifact in the BizTalk interface:

That is to say:

***TargetNameSpace#RootName***

**Creating the Send Port:**

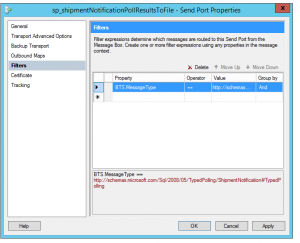
From the BizTalk Administrative tool and in the newly deployed application let’s create a new one way send port and with a filter on BizTalk Message type as follows:

[](http://blog.tallan.com/wp-content/uploads/2014/12/CreatingTheSendPort.png)

Filters

Setting the Send Port Filter to create the subscriber

Here is the filter required:

[](http://blog.tallan.com/wp-content/uploads/2014/12/Filters.png)

After creating a send port that grabs all records from this new receive location, Start the application. When we put a new row into my database we should get one file emitted to disk.

If we create three records in the database, then we still get a single message/file out of BizTalk.

[](http://blog.tallan.com/wp-content/uploads/2014/12/SingleFileOut.png)

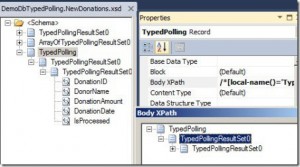
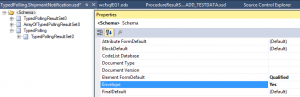
**Debatching**:

The final step in this exercise is optional and only need to be completed if we want to split this up so that these three records show up as three distinct messages/files

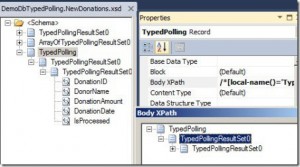
With the WCF adapter, you’ll see that we actually have three root nodes as part of the generated schema.  We can confirm this by looking at the **Schemas** section in the BizTalk Administration Console:

[Debatching](http://blog.tallan.com/wp-content/uploads/2014/12/Debatching.png)

So, this means that we should be able to change the existing schema to support ***debatching***, and it all just works.  Let’s try that now:  Go back to the auto-generated schema, click the topmost **Schema** node, and changed its Envelope property to **Yes**.

[](http://blog.tallan.com/wp-content/uploads/2014/12/BodyXPath.jpg) [](http://blog.tallan.com/wp-content/uploads/2014/12/EnvelopePropertyYES.png)

Next, click the **TypedPolling** node (which acts as the primary root that comes out of the adapter) and set the **Body XPath** value to the node ABOVE the eventual leaf node:

[](http://blog.tallan.com/wp-content/uploads/2014/12/BodyXPath.jpg)

Finally, select the leaf node and set its **Max Occurrence** from **Unbounded** to **1. Rebuild the** project and then redeployed it to the BizTalk Server.  When we add two records to the database, we end up with two records out on disk.

When the record gets debatched automatically by BizTalk in the XML receive pipeline, the resulting **TypedPollingResultSet0** message(which matches a message type known by BizTalk) gets put in the MessageBox and routed around.

Good luck and Happy BizTalking!