

Database Synchronization with Microsoft Sync Framework

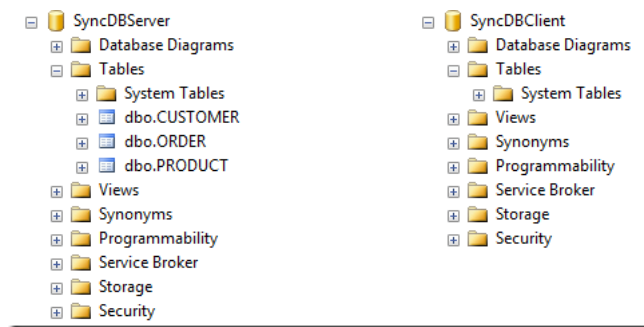
Microsoft Sync Framework is a comprehensive synchronization platform enabling collaboration and offline for applications, services and devices. This is actually shipped with Visual Studio 2008 for the first time and now the current stable version is [Microsoft Sync Framework 2.1](#). There are number of providers in Sync Framework which supports many common data sources. The following are the providers included,

- Database synchronization providers: Synchronization for ADO.NET-enabled data sources.
- File synchronization provider: Synchronization for files and folders.
- Web synchronization components: Synchronization for FeedSync feeds such as RSS and ATOM feeds.

I am not going to write much intro about Microsoft Sync Framework as you can find many articles on internet. Let's see how database synchronization works with Microsoft Sync Framework by a simple example.

I have two Microsoft SQL Server 2008R2 Databases which are "SyncDBServer" and "SyncDBClient". Unfortunately I am having them on the same server and actually what I want to demonstrate is data synchronization between two remote database servers. Since I don't have such environment I am using the same server, because it will not do any harm to the logic I am using here.

In "SyncDBServer", I have three tables which are CUSTOMER, PRODUCT and ORDER. I have filled up these tables with sample data and "SyncDBClient" database has no tables. What I am going to do today is synchronize only the CUSTOMER and PRODUCT tables of "SyncDBServer" with "SyncDBClient". In this case Microsoft Sync Framework will create CUSTOMER and PRODUCT tables in "SyncDBClient" for me.



"SyncDBServer" and "SyncDBClient"

First I need to prepare or provision these two databases for the synchronization. For that I will have to write some codes. Let's create a console application for provisioning "SyncDBServer" and I am naming it as "ProvisionServer". I am adding following references (I am using [Microsoft Sync Framework 2.1 Software Development Kit \(SDK\)](#)).

- Microsoft.Synchronization.Data : Version 3.1.0.0
- Microsoft.Synchronization.Data.SqlServer: Version 3.1.0.0

```
using System.Data.SqlClient;
using Microsoft.Synchronization.Data;
using Microsoft.Synchronization.Data.SqlServer;

namespace ProvisionServer
{
    class Program
    {
        static void Main(string[] args)
        {
            // connect to server database
            SqlConnection serverConn = new SqlConnection("Data Source=.; Initial Catalog=SyncDBServer; Integrated Security=True");

            // define a new scope named MySyncScope
            DbSyncScopeDescription scopeDesc = new DbSyncScopeDescription("MySyncScope");

            // get the description of the CUSTOMER & PRODUCT table from SERVER database
            DbSyncTableDescription cusTableDesc = SqlSyncDescriptionBuilder.GetDescriptionForTable("CUSTOMER", serverConn);
            DbSyncTableDescription prodTableDesc = SqlSyncDescriptionBuilder.GetDescriptionForTable("PRODUCT", serverConn);

            // add the table description to the sync scope definition
            scopeDesc.Tables.Add(cusTableDesc);
            scopeDesc.Tables.Add(prodTableDesc);

            // create a server scope provisioning object based on the MySyncScope
            SqlSyncScopeProvisioning serverProvision = new SqlSyncScopeProvisioning(serverConn, scopeDesc);
```

```

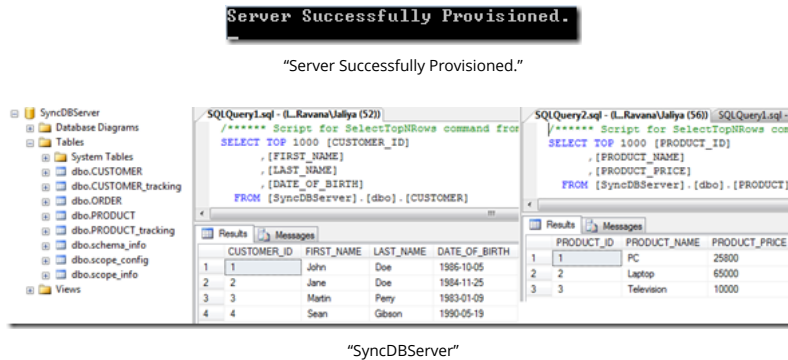
// skipping the creation of table since table already exists on server
serverProvision.SetCreateTableDefault(DbSyncCreationOption.Skip);

// start the provisioning process
serverProvision.Apply();

Console.WriteLine("Server Successfully Provisioned.");
Console.ReadLine();
}
}
}

```

Here I am accessing “SyncDBServer” database and specified my synchronization scope. Now when I run this project, I can see the message “Server Successfully Provisioned.” and most importantly I can see some new tables created in “SyncDBServer” database.



Now I am creating a console application for provisioning “SyncDBClient” and I am naming it as “ProvisionClient”. Again I am adding the same references.

```

using System;
using System.Data.SqlClient;
using Microsoft.Synchronization.Data;
using Microsoft.Synchronization.Data.SqlServer;

namespace ProvisionClient
{
    class Program
    {
        static void Main(string[] args)
        {
            // create a connection to the client database
            SqlConnection clientConn = new SqlConnection(@"Data Source=.; Initial Catalog=SyncDBClient; Integrated Security=True");

            // create a connection to the server database
            SqlConnection serverConn = new SqlConnection(@"Data Source=.; Initial Catalog=SyncDBServer; Integrated Security=True");

            // get the description of SyncScope from the server database
            DbSyncScopeDescription scopeDesc = SqlSyncDescriptionBuilder.GetDescriptionForScope("MySyncScope", serverConn);

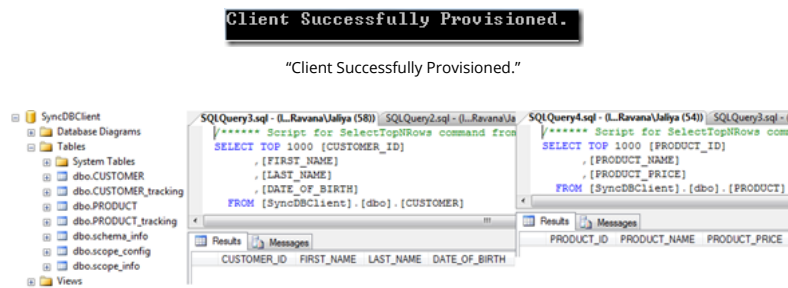
            // create server provisioning object based on the SyncScope
            SqlSyncScopeProvisioning clientProvision = new SqlSyncScopeProvisioning(clientConn, scopeDesc);

            // starts the provisioning process
            clientProvision.Apply();

            Console.WriteLine("Client Successfully Provisioned.");
            Console.ReadLine();
        }
    }
}

```

Here I am accessing both “SyncDBServer” and “SyncDBClient”. I am getting the synchronization scope from “SyncDBServer” and applying it on the “SyncDBClient”. Now when I run this project, I can see the message “Client Successfully Provisioned.” and here also I can see some new tables created in “SyncDBClient” database including CUSTOMER and PRODUCT tables. But there will be no data in these two tables.



Now finally what's left to do is the synchronization. For that I am creating a console application and I am naming it as "ExecuteSync". I am adding following references.

- Microsoft.Synchronization: Version 2.1.0.0
- Microsoft.Synchronization.Data : Version 3.1.0.0
- Microsoft.Synchronization.Data.SqlServer: Version 3.1.0.0

```
using System;
using System.Data.SqlClient;
using Microsoft.Synchronization;
using Microsoft.Synchronization.Data;
using Microsoft.Synchronization.Data.SqlServer;

namespace ExecuteSync
{
    class Program
    {
        static void Main(string[] args)
        {
            SqlConnection clientConn = new SqlConnection(@"Data Source=.; Initial Catalog=SyncDBClient; Integrated Security=True");

            SqlConnection serverConn = new SqlConnection("Data Source=.; Initial Catalog=SyncDBServer; Integrated Security=True");

            // create the sync orchestrator
            SyncOrchestrator syncOrchestrator = new SyncOrchestrator();

            // set local provider of orchestrator to a sync provider associated with the
            // MySyncScope in the client database
            syncOrchestrator.LocalProvider = new SqlSyncProvider("MySyncScope", clientConn);

            // set the remote provider of orchestrator to a server sync provider associated with
            // the MySyncScope in the server database
            syncOrchestrator.RemoteProvider = new SqlSyncProvider("MySyncScope", serverConn);

            // set the direction of sync session to Upload and Download
            syncOrchestrator.Direction = SyncDirectionOrder.UploadAndDownload;

            // subscribe for errors that occur when applying changes to the client
            ((SqlSyncProvider)syncOrchestrator.LocalProvider).ApplyChangeFailed += new EventHandler<DbApplyChangeFailedEventArgs>(Program_ApplyChangeFailed);

            // execute the synchronization process
            SyncOperationStatistics syncStats = syncOrchestrator.Synchronize();

            // print statistics
            Console.WriteLine("Start Time: " + syncStats.SyncStartTime);
            Console.WriteLine("Total Changes Uploaded: " + syncStats.UploadChangesTotal);
            Console.WriteLine("Total Changes Downloaded: " + syncStats.DownloadChangesTotal);
            Console.WriteLine("Complete Time: " + syncStats.SyncEndTime);
            Console.WriteLine(String.Empty);
            Console.ReadLine();
        }

        static void Program_ApplyChangeFailed(object sender, DbApplyChangeFailedEventArgs e)
        {
            // display conflict type
            Console.WriteLine(e.Conflict.Type);

            // display error message
            Console.WriteLine(e.Error);
        }
    }
}
```

Here I am mentioning the synchronization direction(Upload or Download or UploadAndDownload or DownloadAndUpload) and doing the synchronization according to my specified scope. When I run this, I am getting the following output.

```
Start Time: 1/31/2013 11:52:01 PM
Total Changes Uploaded: 0
Total Changes Downloaded: 7
Complete Time: 1/31/2013 11:52:13 PM
```

Synchronization Completed.

CUSTOMER_ID	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH
1	John	Doe	1986-10-05
2	Jane	Doe	1984-11-25
3	Martin	Perry	1983-01-09
4	Sean	Gibson	1990-05-19

PRODUCT_ID	PRODUCT_NAME	PRODUCT_PRICE
1	PC	25800
2	Laptop	65000
3	Television	10000


"SyncDBClient" after Synchronization

That's it. I am uploading sample databases and project files to my SkyDrive. You can download and play around.

DB-DB

Happy Coding.

Regards,
Jaliya

Posted by jaliya udagedara at 12:21 AM 

 +5 Recommend this on Google

Reactions: interesting (0) cool (1)

9 comments:



Witman Gonzalez August 17, 2013 at 3:45 AM

How do I make the derpvosion

[Reply](#)



Witman Gonzalez August 17, 2013 at 3:46 AM

I did everything but does not sync when I need to update values on fields

[Reply](#)

Anonymous October 22, 2013 at 8:40 AM

do you have a syncprovider for mysql?

[Reply](#)



Vijaykumar December 18, 2013 at 1:33 PM

Hi I have referred to your article and created a POC and works fine for me.
All I want is some easy solution to filter the records while inserting the records to the client as per the my input
e.g. client A want records only related to Client A and not Client B, C or etc.
How can I do achieve this?

[Reply](#)

[Replies](#)



Jaliya Udagedara December 18, 2013 at 1:43 PM

Refer this please,

<http://jaliyaudagedara.blogspot.com/2013/02/synchronize-with-parameter-based.html>

[Reply](#)



Yusuf Karatoprak April 23, 2014 at 7:09 PM

i have really strange problem. i have been trying to synchronize 2 tables but they are localized in same db. it is easy if you have 2 tables and 2 database (ClentDb and ServerDb). i dont have clientDb. but i have 2 tables. if you have 2 tables(they should be compared) in same database,Sync is not working in the same db. how can i solve this problem?

[Reply](#)

Anonymous October 23, 2014 at 1:17 PM

How to use this code in WCF?

i want to sync the server and client in the exact way that this console application is sync but not with this console application i want to sync with WCF

[Reply](#)



Cong Thanh Ho Nguyen November 6, 2015 at 10:39 AM

Good articles, thank you :)

[Reply](#)

Anonymous November 24, 2015 at 8:50 PM

Thanks for the article, just what I was looking for.
I translated it into vb.net and got 2 errors:

1. Compile error

```
DirectCast(syncOrchestrator.LocalProvider, SqlSyncProvider).ApplyChangeFailed += New EventHandler(Of DbApplyChangeFailedEventArgs)(AddressOf Program_ApplyChangeFailed)
```

Error message"

Public Event ApplyChangeFailed(sender As Object, e As Microsoft.Synchronization.Data.DbApplyChangeFailedEventArgs) is an event, and cannot be called directly. Use a 'RaiseEvent' statement to raise an event.

2. Sync problem

```
Dim syncStats As SyncOperationStatistics = syncOrchestrator.Synchronize()
```

Results in:

Cannot add the SyncAdapter to the SyncAdapterCollection. A SyncAdapter for table 'BrandFreqConverter' already exists.

Can you tell what i'm doing wrong?

[Reply](#)