# How to Execute the SqlCommand with a SqlNotificationRequest

## Introduction

After executing a command on database to get data, the data may be changed the other clients. If the application need the latest data, it need a notification from the server. In this application, we will demonstrate how to execute the SqlCommand with a SqlNotificationRequest:

1. Set and execute the SqlCommand with a SqlNotificationRequest;

2. Begin to monitor the queue in SqlServer;

3. Refresh the data.

## Building the Sample

Before you run the sample, you need to finish the following steps:

Step1. Please choose one of the following ways to build the database:

* Attach the database file MySchool.mdf under the folder \_External\_Dependecies to your SQL Server (2008 or later version) database instance. After attaching, please execute the following script in the SqlServer:

ALTER DATABASE MySchool SET ENABLE\_BROKER;

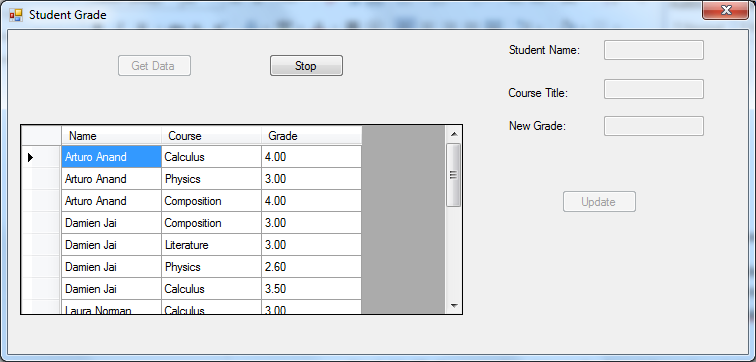
* Run the MySchool.sql script under the folder \_External\_Dependecies in your SQL Server (2008 or later version) database instance.

Step2. Modify the connection string in the Project Properties->Settings according-> MySchoolConnectionString to your SQL Server (2008 or later version) database instance name

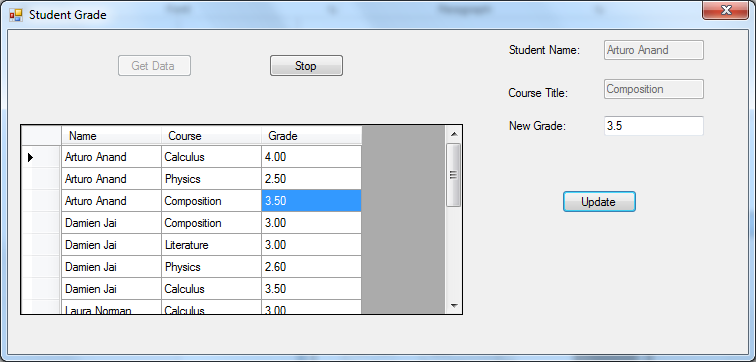
## Running the Sample

Press F5 to run the sample.

1. Click the **Get Data** button to get all the data:



2. You can click the cells in the DataGridView to choose a row, and then you can input a new grade. After click the **Update** button, the value in DataGridView will be updated.



3. You can click **Stop** button to stop the monitoring.

## Using the Code

1. Register SqlNotificationRequest

Before we use the SqlNotificationRequest, we need to create it.

|  |
| --- |
| -Code block start-  --C# code snippet start--  request = new SqlNotificationRequest();  request.UserData = new Guid().ToString();  request.Options = String.Format("Service={0};", serviceName);  request.Timeout = notificationTimeout;  --C# code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

2. Create the Command and bind the SqlNotificationRequest

We create a command to fill the data and bind the SqlNotificationRequest to the command.

|  |
| --- |
| -Code block start-  --C# code snippet start--  command.Notification = null;  command.Notification = notification.NotificationRequest;  using (SqlDataAdapter adapter = new SqlDataAdapter(command))  {  adapter.Fill(dataToWatch, tableName);  dgvWatch.DataSource = dataToWatch;  dgvWatch.DataMember = tableName;  }  --C# code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

3. Begin to Monitor the Service Broker queue

We open a new thread to monitor the Service Broker queue.

|  |
| --- |
| -Code block start-  --C# code snippet start--  public void StartSqlNotification()  {  listenTask = new Task(Listen);  listenTask.Start();  }  --C# code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

4. Monitoring the Service Broker queue

If we get a new message from the queue, we will refresh the data.

|  |
| --- |
| -Code block start-  --C# code snippet start--  private void Listen()  {  using (SqlConnection conn = new SqlConnection(connectionString))  {  using (cmd = new SqlCommand(listenSql, conn))  {  if (conn.State != ConnectionState.Open)  {  conn.Open();  }  cmd.CommandTimeout = notificationTimeout + 120;  using (SqlDataReader reader = cmd.ExecuteReader())  {  while (reader.Read())  {  for (int i = 0; i <= reader.FieldCount - 1; i++)  Debug.WriteLine(reader[i].ToString());  }  }  }  }  RegisterSqlNotificationRequest();  }  --C# code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

5. Invoke the OnChanged event

If the data was changed and the OnChanged event isn’t null, this event will be invoked.

|  |
| --- |
| -Code block start-  --C# code snippet start--  if (OnChanged != null)  {  OnChanged(this, null);  }  --C# code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

## More Information

[SqlCommand Execution with a SqlNotificationRequest](http://msdn.microsoft.com/en-us/library/wd2x83zk(VS.110).aspx)

[Enabling Query Notifications](http://msdn.microsoft.com/en-us/library/ms172133.aspx)

[SqlNotificationRequest Constructor](http://msdn.microsoft.com/query/dev11.query?appId=Dev11IDEF1&l=EN-US&k=k(%22System.Data.Sql.SqlNotificationRequest.%23ctor%22);k(TargetFrameworkMoniker-.NETFramework,Version%3Dv4.5);k(DevLang-csharp)&rd=true)