**In this video we will discuss the use**  
**1.** BatchSize property  
**2.** NotifyAfter property  
**3.** SqlRowsCopied event   
  
   
  
**BatchSize property** - Specifies the number of rows in a batch that will be copied to the destination table. The BatchSize property is very important as the performance of data transfer depends on it. The default batch size is 1. In the example below, BatchSize is set to 10000. This means once the reader has read 10000 rows they will be sent to the database as a single batch to perform the bulk copy operation.  
  
**NotifyAfter property** - Defines the number of rows to be processed before raising SqlRowsCopied event. In the example below, NotifyAfter property is set to 5000. This means once every 5000 rows are copied to the destination table SqlRowsCopied event is raised.  
  
**SqlRowsCopied event** - This event is raised every time the number of rows specified by NotifyAfter property are processed. This event is useful for reporting the progress of the data transfer.  
  
Let us now understand these properties with an example.  
  
**Step 1 :**Execute the following SQL script to create **Products\_Source**table and populate it with test data.

Create Table Products\_Source

(

 [Id] int primary key,

 [Name] nvarchar(50),

 [Description] nvarchar(250)

)

GO

Declare @Id int

Set @Id = 1

While(@Id <= 300000)

Begin

 Insert into Products\_Source values

 (@Id, 'Product - ' + CAST(@Id as nvarchar(20)),

 'Product - ' + CAST(@Id as nvarchar(20)) + ' Description')

 Print @Id

 Set @Id = @Id + 1

End

GO

**Step 2 :** Create **Products\_Destination**table

Create Table Products\_Destination

(

 [Id] int primary key,

 [Name] nvarchar(50),

 [Description] nvarchar(250)

)

GO

**Step 3 :** Create a new console application. Name it Demo. Include the database connection string in App.config file

<connectionStrings>

  <add name="CS"

        connectionString="server=.;database=Sample;integrated security=SSPI"/>

</connectionStrings>

**Step 4 :** Copy and paste the following code in **Program.cs**file.

using System;

using System.Configuration;

using System.Data.SqlClient;

namespace Demo

{

    class Program

    {

        static void Main()

        {

            string cs = ConfigurationManager.ConnectionStrings["CS"].ConnectionString;

            using (SqlConnection sourceCon = new SqlConnection(cs))

            {

                SqlCommand cmd = new

                                       SqlCommand("Select \* from Products\_Source", sourceCon);

                sourceCon.Open();

                using (SqlDataReader rdr = cmd.ExecuteReader())

                {

                    using (SqlConnection destinationCon = new SqlConnection(cs))

                    {

                        using (SqlBulkCopy bc = new SqlBulkCopy(destinationCon))

                        {

                            bc.BatchSize = 10000;

                            bc.NotifyAfter = 5000;

                            bc.SqlRowsCopied +=

                                     new SqlRowsCopiedEventHandler(bc\_SqlRowsCopied);

                            bc.DestinationTableName = "Products\_Destination";

                            destinationCon.Open();

                            bc.WriteToServer(rdr);

                        }

                    }

                }

            }

        }

        static void bc\_SqlRowsCopied(object sender, SqlRowsCopiedEventArgs e)

        {

            Console.WriteLine(e.RowsCopied + " loaded....");

        }

    }

}

**Please Note :** Instead of creating a separate event handler method for **SqlRowsCopied**event and then passing the name of the event handler method to **SqlRowsCopiedEventHandler**delegate, we can reduce the amount of code we have to write with the help of an anonymous function as shown below.

bc.SqlRowsCopied += (sender, eventArgs) =>

{

    Console.WriteLine(eventArgs.RowsCopied + " loaded....");

};

**This means the following method is no longer required**

static void bc\_SqlRowsCopied(object sender, SqlRowsCopiedEventArgs e)

{

    Console.WriteLine(e.RowsCopied + " loaded....");

}