

Student Data Aggregation Calculator

by



Administrator Guide

Version: 1.0

Date: February 21, 2013

Table of Contents

Introduction.....	3
About this Guide.....	3
System Operating Environment.....	3
Database.....	5
Restore Database.....	5
Create Database Backup of SDAC Database.....	14
Run the Database Script on SDAC database.....	19
Internet Information Services (IIS).....	25
Install Microsoft .NET Framework 4.0 Setup.....	25
Publish SDAC in IIS 7.....	27
Un-Installation	31
Un-Install Microsoft .NET Framework 4.0.....	31
Register SDAC.....	34
Steps for Application Registration.....	34
The Web.config file.....	38
Database Connections.....	42

Introduction

The Student Data Aggregation Calculator (SDAC) is a web-based application that allows district personnel the ability to view student data that is both relevant to their needs and that can lead to some actionable outcome.

inBloom through its shared technology infrastructure is opening doors for school district staff that have typically only interacted with software systems as a means for entering student data or viewing static reports that are often outdated and incomplete. Given the breadth and depth of data the inBloom Data Store encompasses, access to real-time, accurate and relevant student data is now possible.

About this Guide

This SDAC Administrator Guide provides the necessary guidelines and instructions for installing and maintaining the application. As a system administrator, you can use this guide as a source of information and solutions for deploying the system. This guide includes the technical details, procedures, and recommendations for successfully installing and maintaining the SDAC.

System Operating Environment

This section lists the hardware and software tools needed to operate and maintain the SDAC on both the client and server side.

Client Environment

The Client Operating Environment includes the hardware and software requirements for machines that access the system for read or write purposes.

Hardware Environment

Although there are no specific hardware requirements on the client, for performance purpose, we recommend using a PC with a minimum CPU speed of 2.6GHz or higher and 1GB or higher memory for better performance. The SDAC is also capable of running on tablet devices such as the iPad.

Software Environment

- Web Browser: Internet Explorer 8.0 / 9.0 Mozilla, Safari, Chrome.
- Operating System: Any OS that supports the above specified browsers.
- Microsoft Office 2007 or later versions.

Host Environment

The Server-Side Operating Environment includes the hardware and software requirements for a machine that runs the system online.

Hardware Environment

N-Tier Architecture with:



- Application Server - because of its unique requirements, which call for the application to support many reporting functions in addition to a large set of transactional features, a minimum of 2GB memory, is recommended to be reserved for the SDAC system for better performance.
- Web Server
- Database Server

Software Environment

- ASP.NET 4.0, Windows Server Hosting, IIS, Web Services Support, Visual Studio Compatible
- Application Server:
 - o Microsoft Office 2010 Web Component (OWC11) with latest security updates
 - o Microsoft .NET Framework 4.0
- Database Management System: Microsoft SQL Server 2008 R2 Enterprise Edition with latest Service Pack

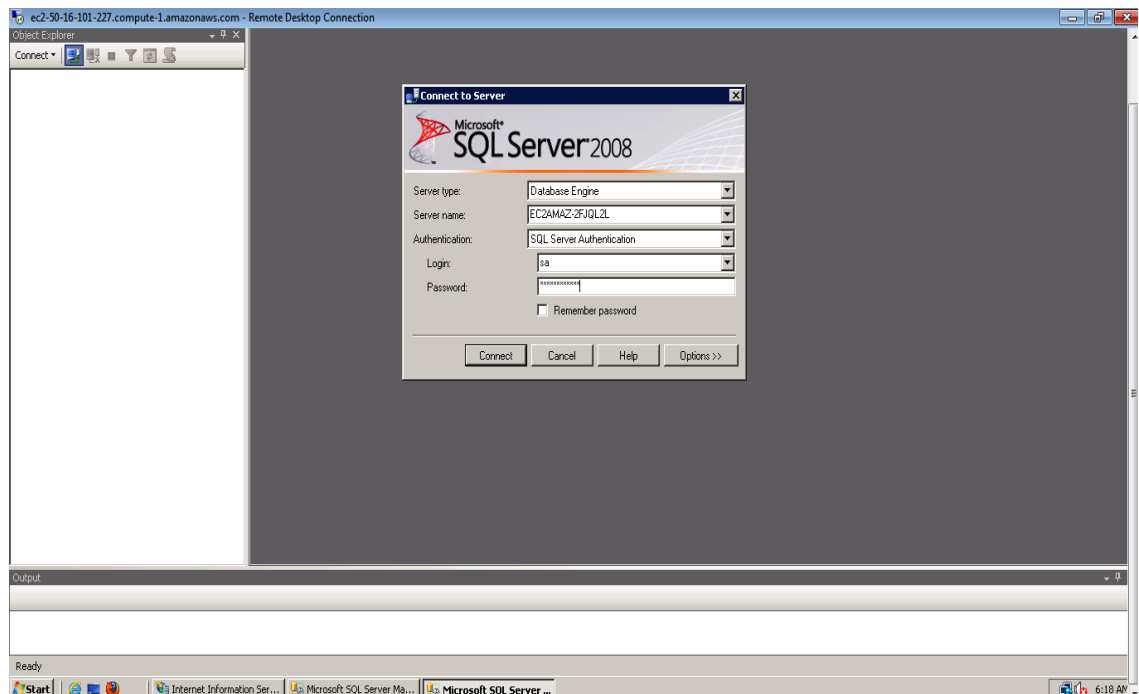
Database

Restore Database

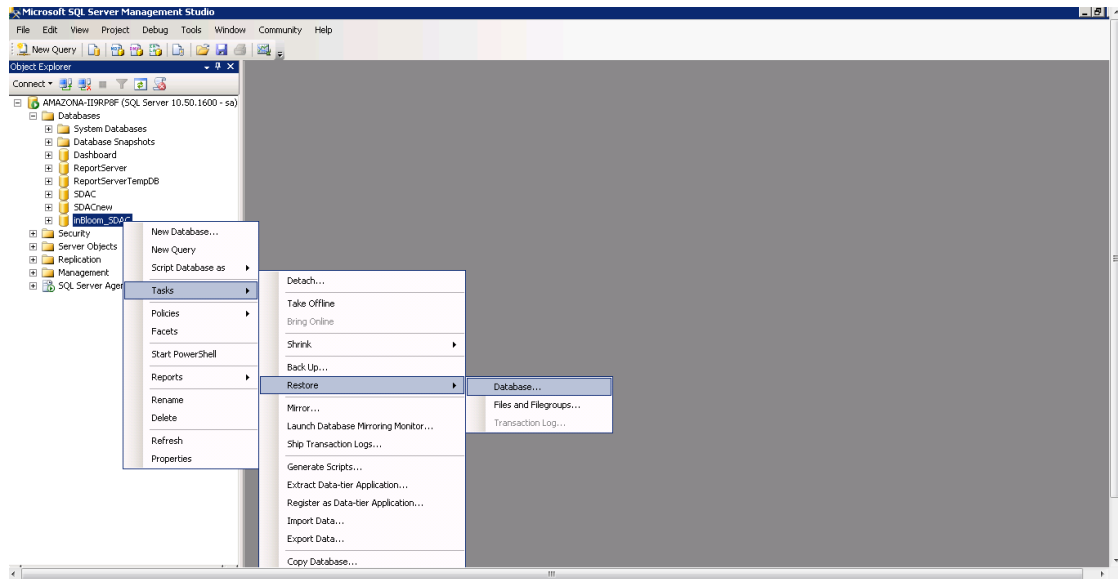
The below section applies only to the Development Environment. Follow the below steps to restore the existing SDAC database with the backup copy provided.


DO NOT Restore an SDAC database in the Production environment.

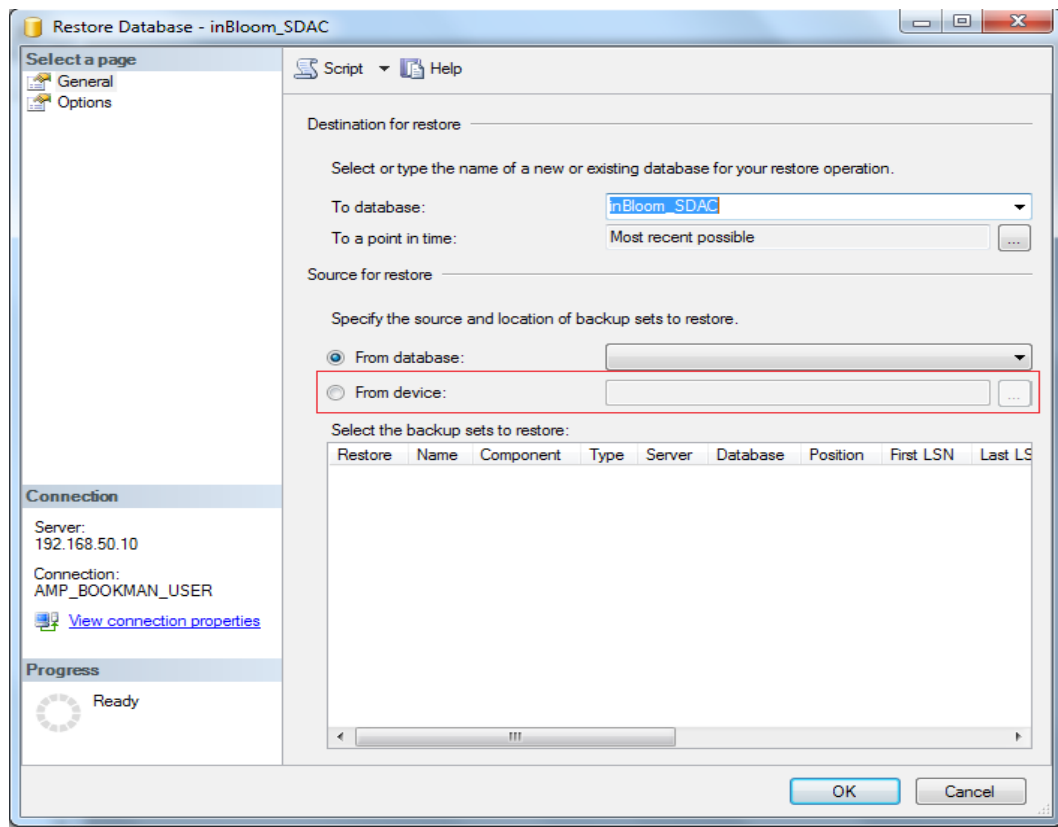
1. Get the database backup from the location provided for SDAC and copy to the local drive of the server. You can delete the database backup file after completing the restore process.
2. Open the **SQL Server Management Studio** from **Start Menu → Programs → Microsoft SQL Server 2008 → SQL Server Management Studio**
3. It will open the Microsoft SQL Server Management Studio window and will prompt for Connect to Server. Select the valid Server Name and Authentication method to login to SQL Server. Click on Connect button to login to SQL Server 2008. See the below figure.



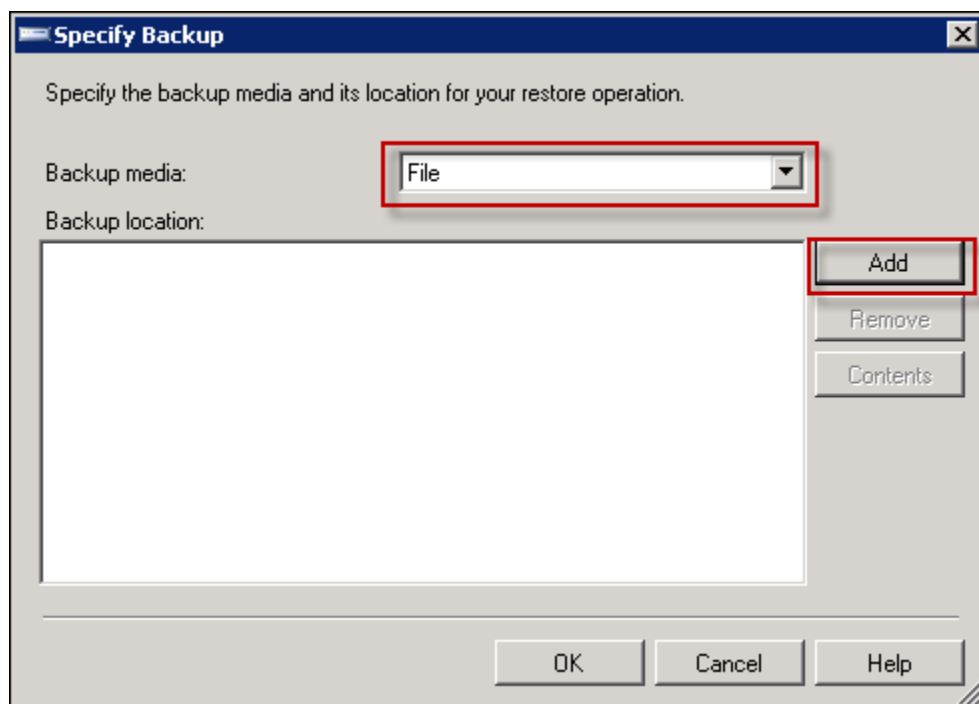
4. After clicking on Connect button you will see the below figure.
5. Right click on **SDAC** Database and select **Tasks** → **Restore** → **Database** (See the below figure)



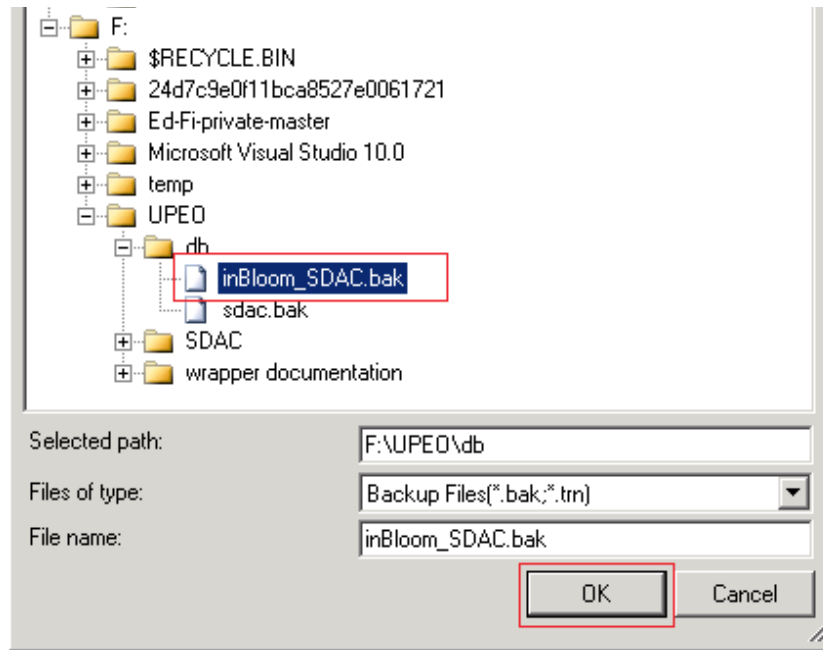
6. This will show you the below figure. Select the **From Device** Option and click on  Button as shown in below figure.



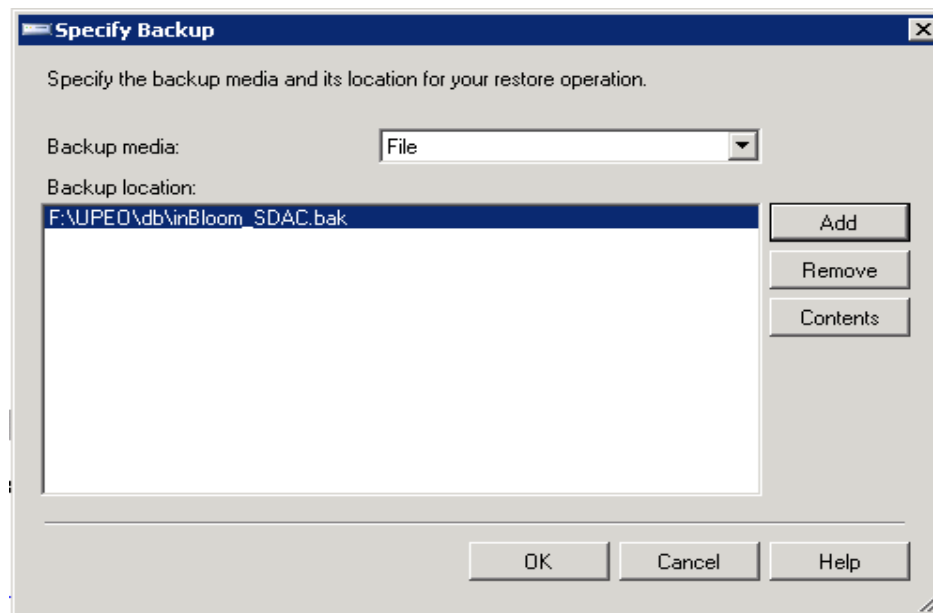
7. This will show you the Specify Backup sub window. Click on Add button to add the backup file to restore. See the below figure.



8. Select the database backup file name from the database backup folder on server's local drive. See the below figure.



9. Click on **OK** button. This will take you back to **Specify Backup** sub window. See the below figure.



10. Click on **OK** button and this will take you back to the **Restore Database – SDAC** window. Select the backup sets to restore as shown in below figure.

Restore Database - inBloom_SDAC

Select a page
 General
 Options

Script Help

Destination for restore

Select or type the name of a new or existing database for your restore operation.

To database: inBloom_SDAC

To a point in time: Most recent possible

Source for restore

Specify the source and location of backup sets to restore.

From database:
 From device: D:\Seabus_Share\raju kurapati\sdac.bak

Select the backup sets to restore:

Restore	Name	Component	Type	Server	Database	Posit
<input checked="" type="checkbox"/>	SDAC-Full Database Backup	Database	Full	SBS-SRV	SDAC	1

Connection

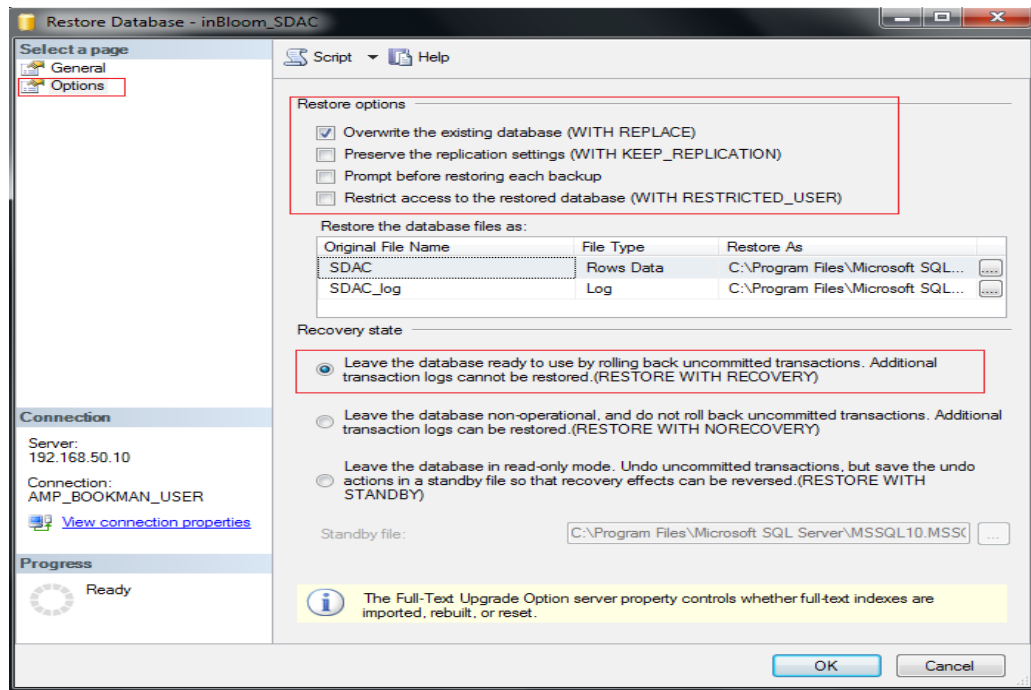
Server: 192.168.50.10
 Connection: AMP_BOOKMAN_USER
[View connection properties](#)

Progress

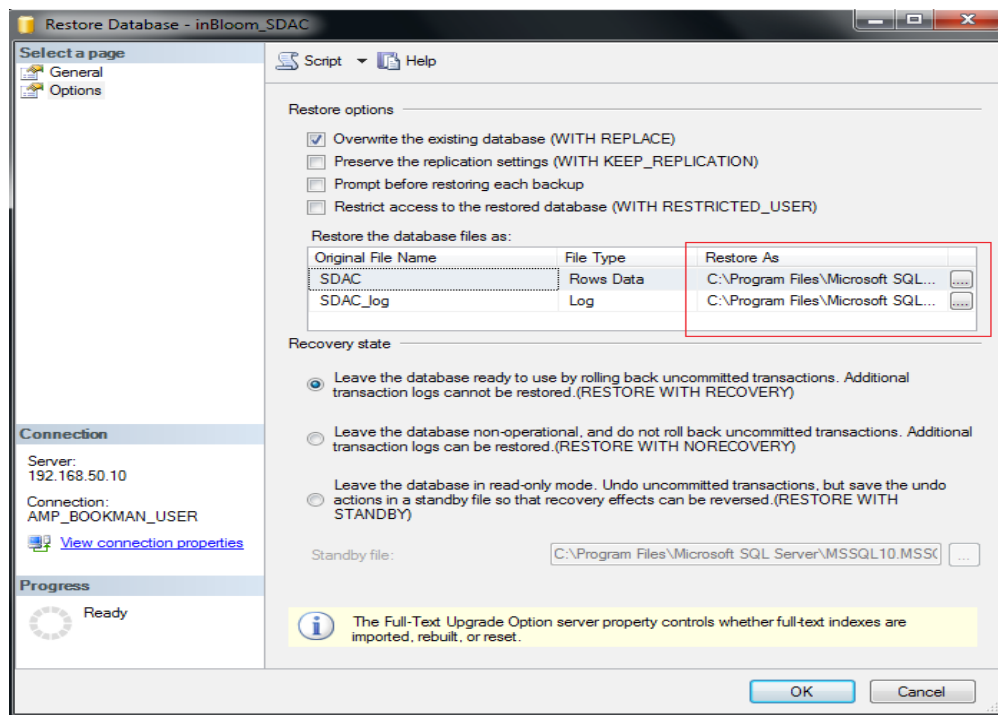
Ready

OK Cancel


11. Select the **Options** from left menu bar then select the Restore option, Recovery State as shown in below figure.



12. Under **Restore the database files as** select the **Restore As** database files (.MDF and .LDF). See the below figure.





13. Select the SDAC.MDF and SDAC_log.LDF location by clicking on  buttons
SDAC file path:

E:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\SDAC.MDF

14. Select the database file name for (.MDF file and .LDF file) and click on **OK** button.

SDAC_log.LDF file path:

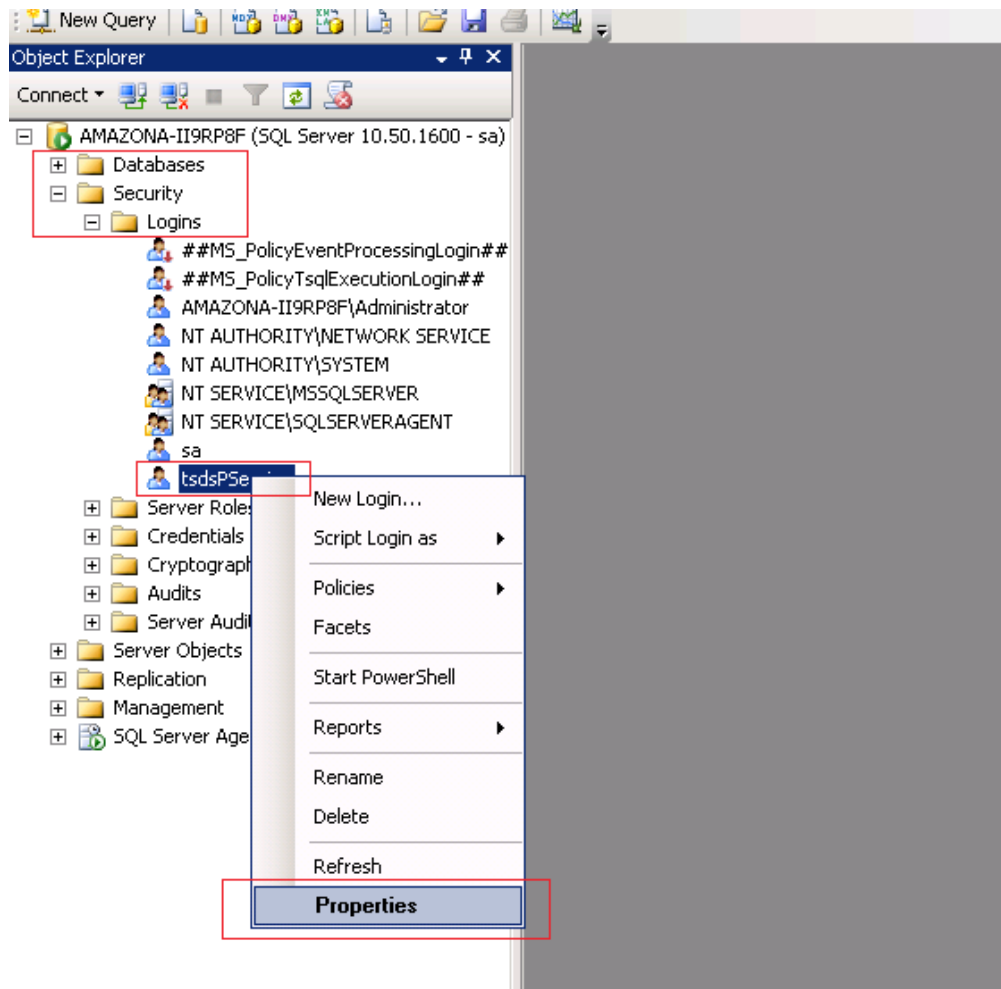
E:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\SDAC.LDF

SDAC_log file path:

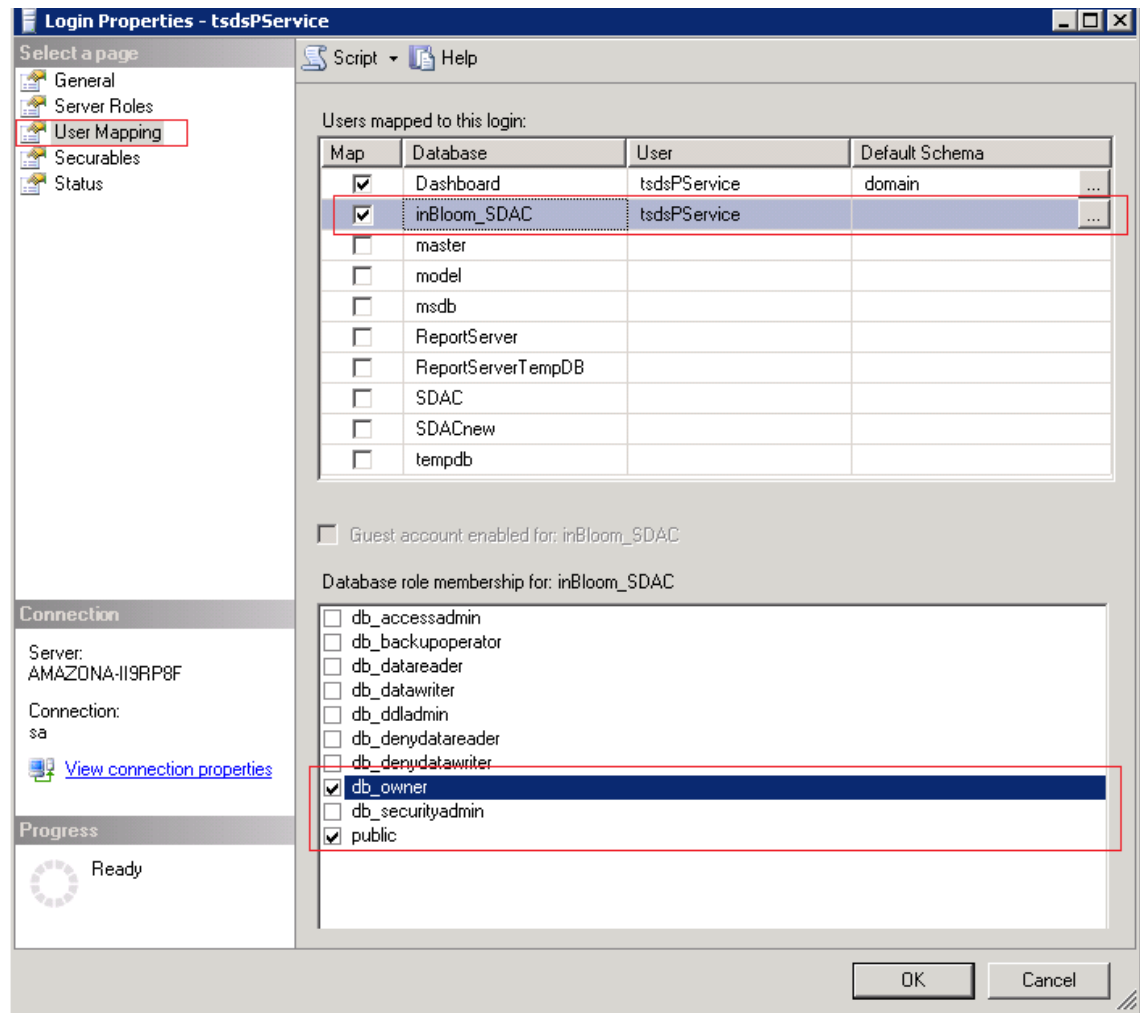
E:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\SDAC_log.LDF

15. After selecting the files (Restore As) you will see the below figure.
16. Click on **OK** button to start restoring the database backup. You will see the below figure after successfully restoring the database backup.
17. You will see the “**The restore of the database ‘SDAC’ completed successfully.**” message. Click on **OK** button to close the message sub window.
18. You have successfully restored the SDAC database with database backup provide.
19. Right Click on **SDAC** user (same in below example) under Logins folder as shown in below figure. Click on Properties option.





20. It will show the below figure. Select **SDAC** database from Users mapped to this login section and select **DB_Owner** under “Database role membership for: **SDAC**”.

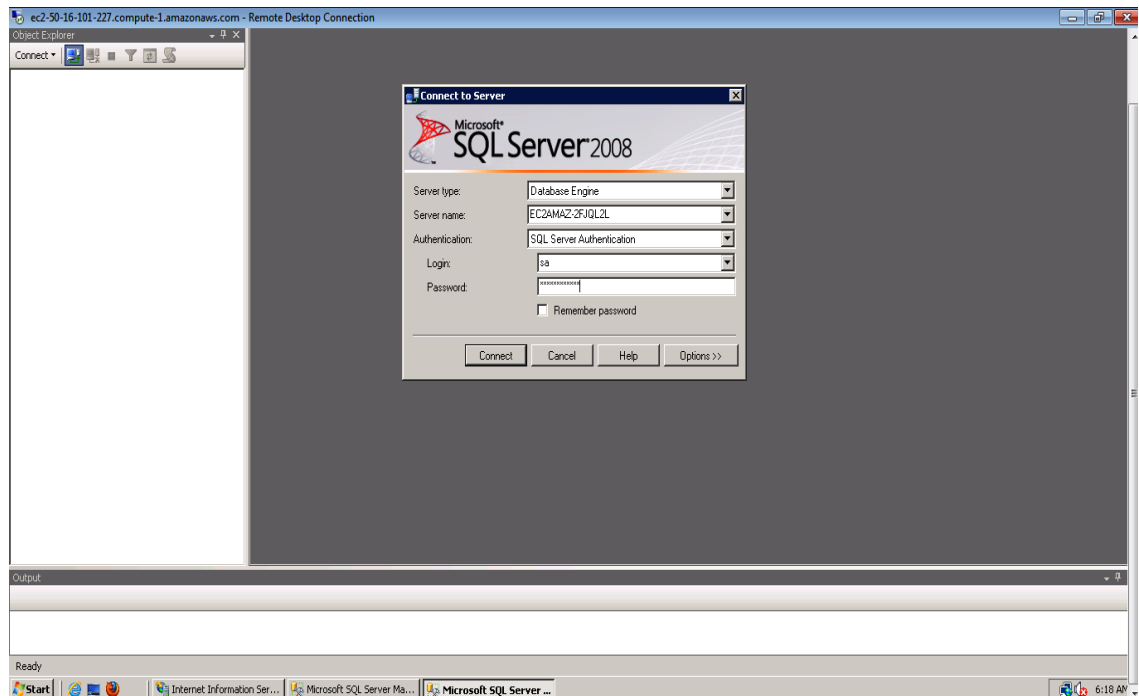


21. Click on OK button. You have successfully granted permission to **SDAC** user for **SDAC** database.

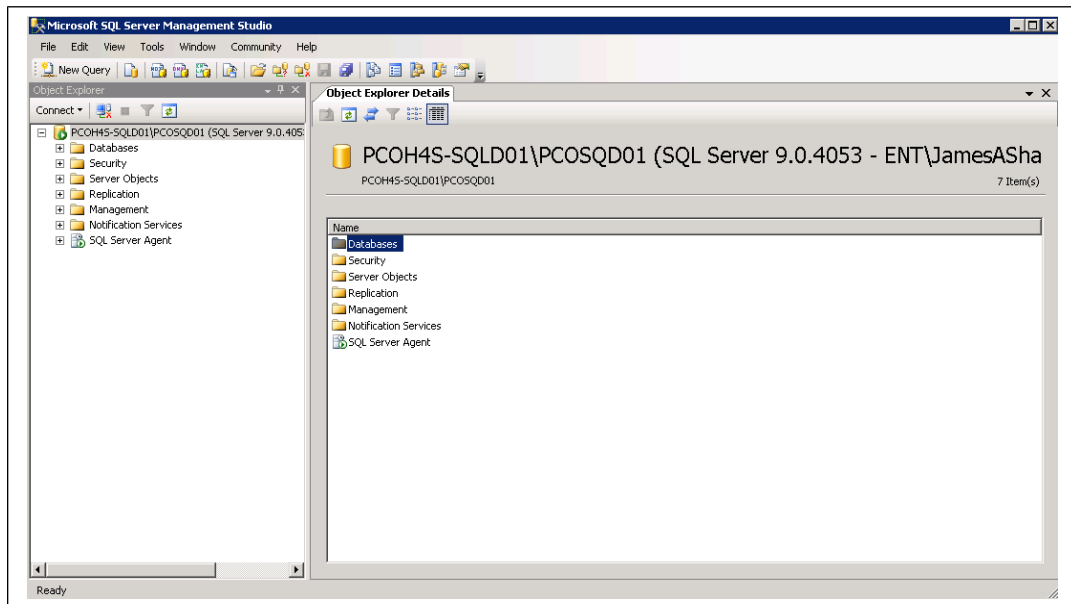
Create Database Backup of SDAC Database

Follow the instructions below to take the database backup of SDAC database before executing the Database Script in next section.

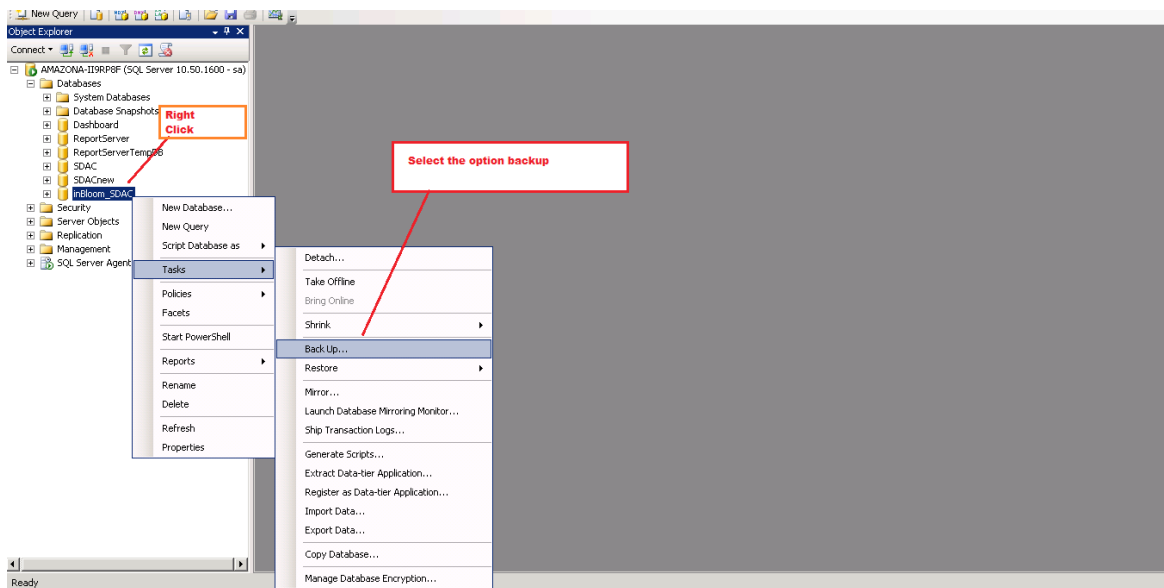
1. Open the **SQL Server Management Studio** from **Start Menu → Programs → Microsoft SQL Server 2008 → SQL Server Management Studio**
2. It will open the Microsoft SQL Server Management Studio window and will prompt for Connect to Server. Select the valid Server Name and Authentication method to login to SQL Server. Click on Connect button to login to SQL Server 2008. See the below figure.



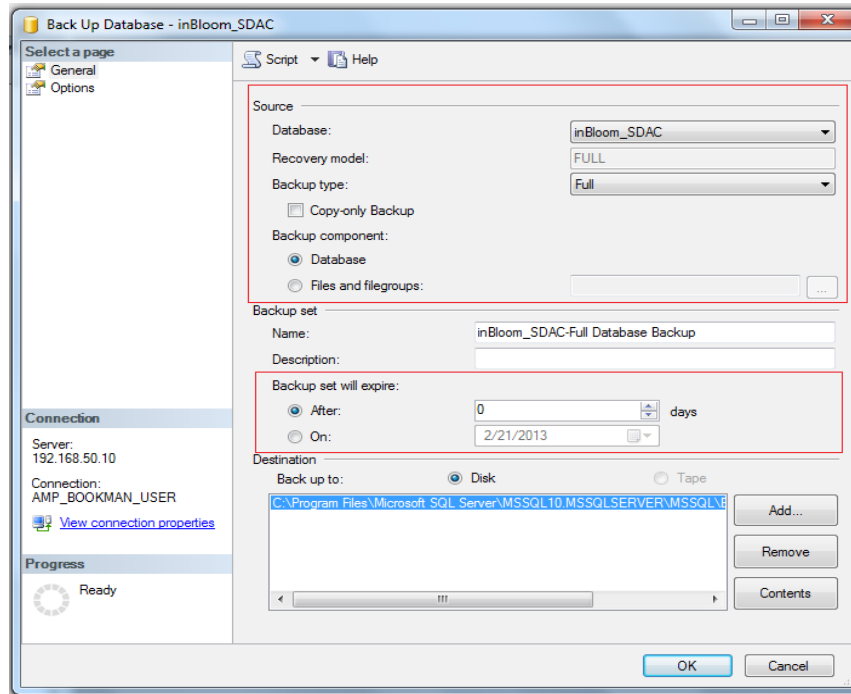
3. After clicking on Connect button you will see the below figure.



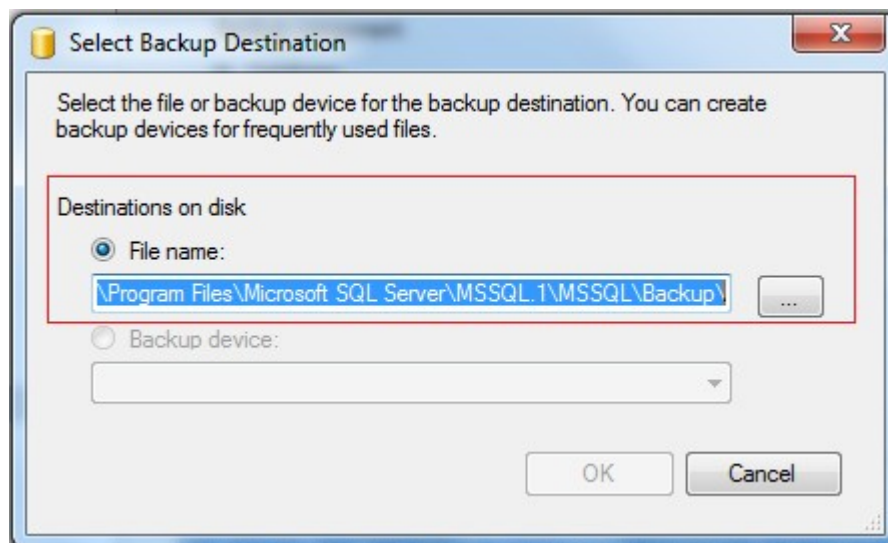
4. Right click on **SDAC** database under databases folder. Select Back Up option from Tasks as show in the below figure.



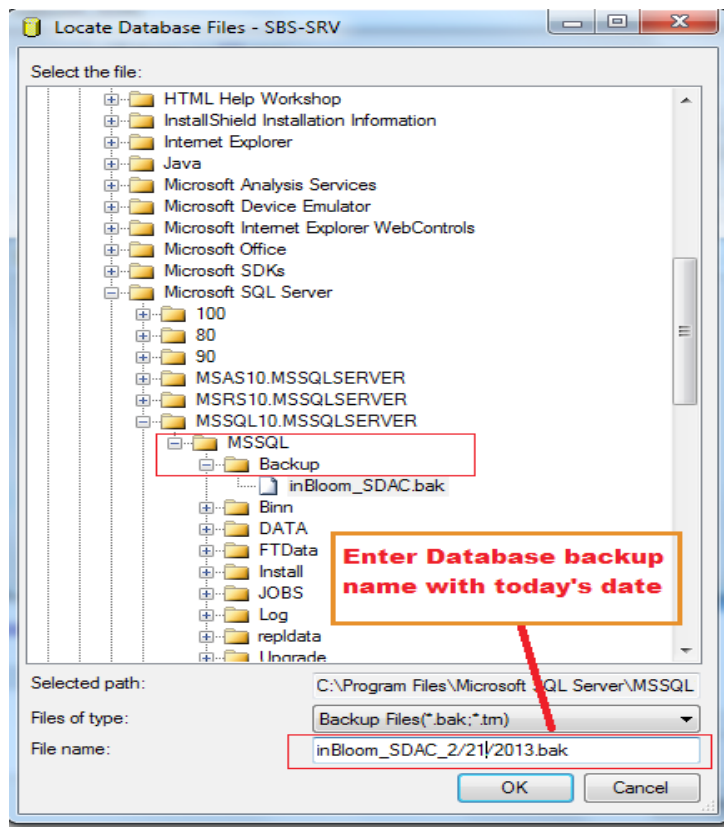
5. After clicking on **Back Up** option, it will display below figure. Make the changes as shown in the below figure and follow next step to change destination to store backup file.



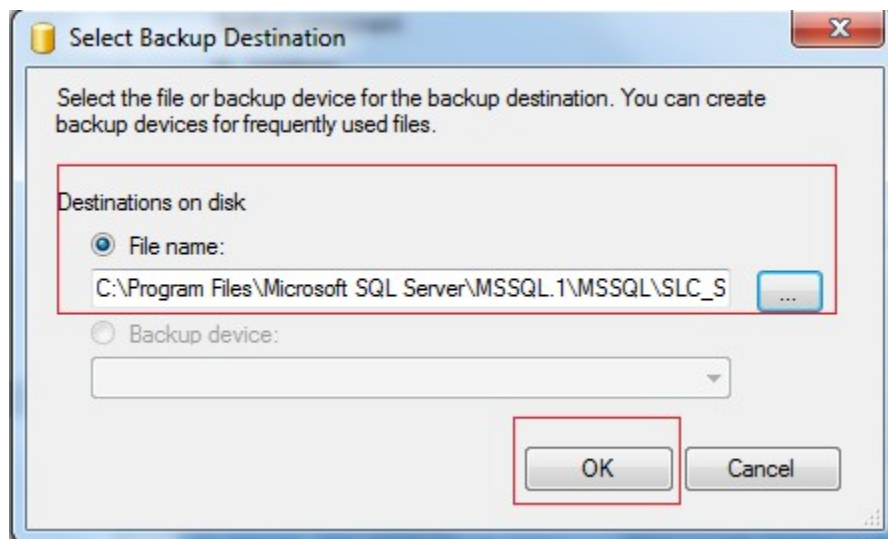
6. Click on **Remove** button if any file exists as show in the above figure.
7. Click on **Add** button and it will display below figure.



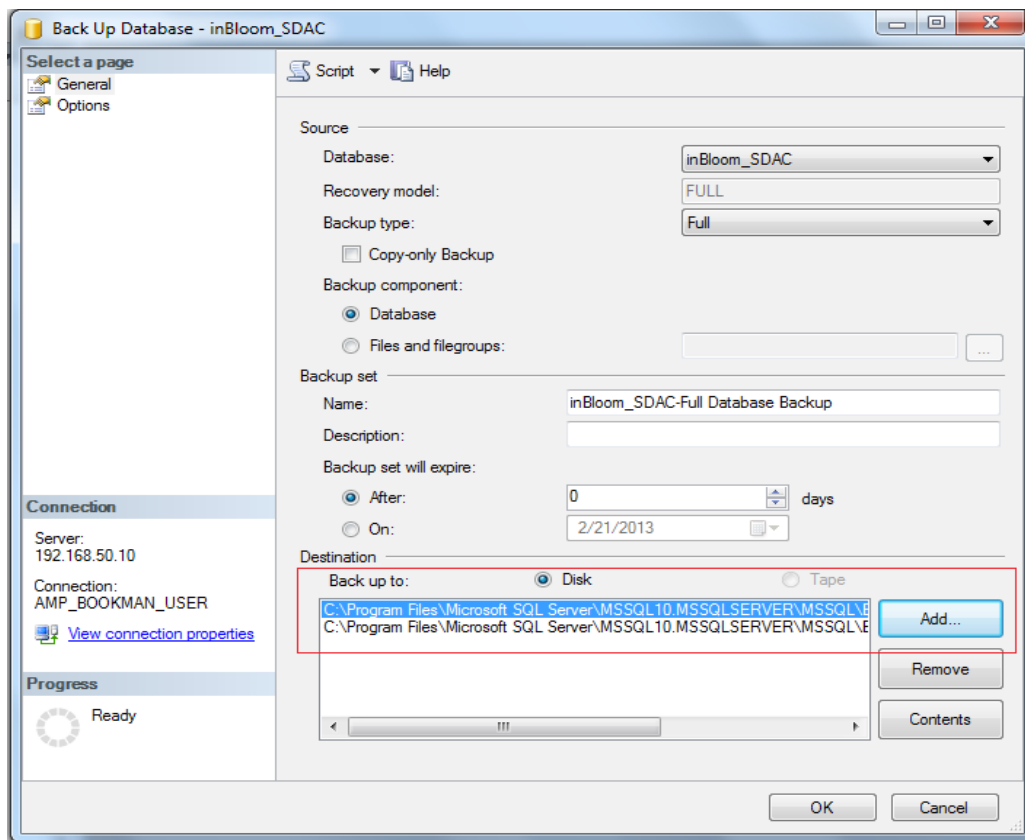
8. Click on **...** button to select Backup filename and location as shown in below figure. You can select any drive (E.g. C: or D:, etc.) to store the backup file.



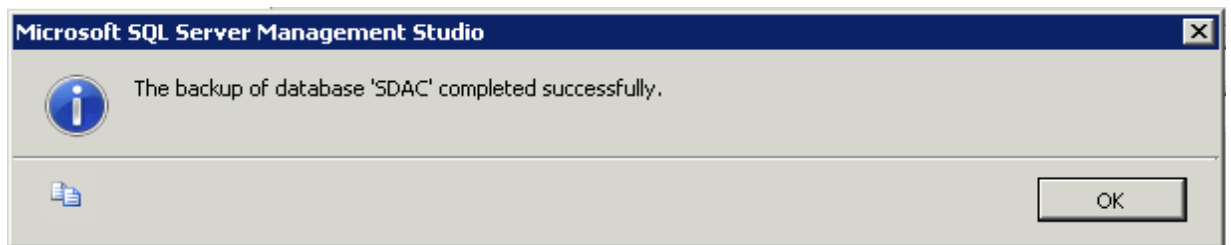
9. Click on **OK** button. It will show you below figure with selected location and filename to store database backup file.



10. Click on **OK** button. It will show you Back Up Database – SDAC screen with database location and filename entered to store the database backup. See below figure.



11. Click on OK button to start backup process. After completing backup, it will show you below figure. Click on OK button to close the message box.



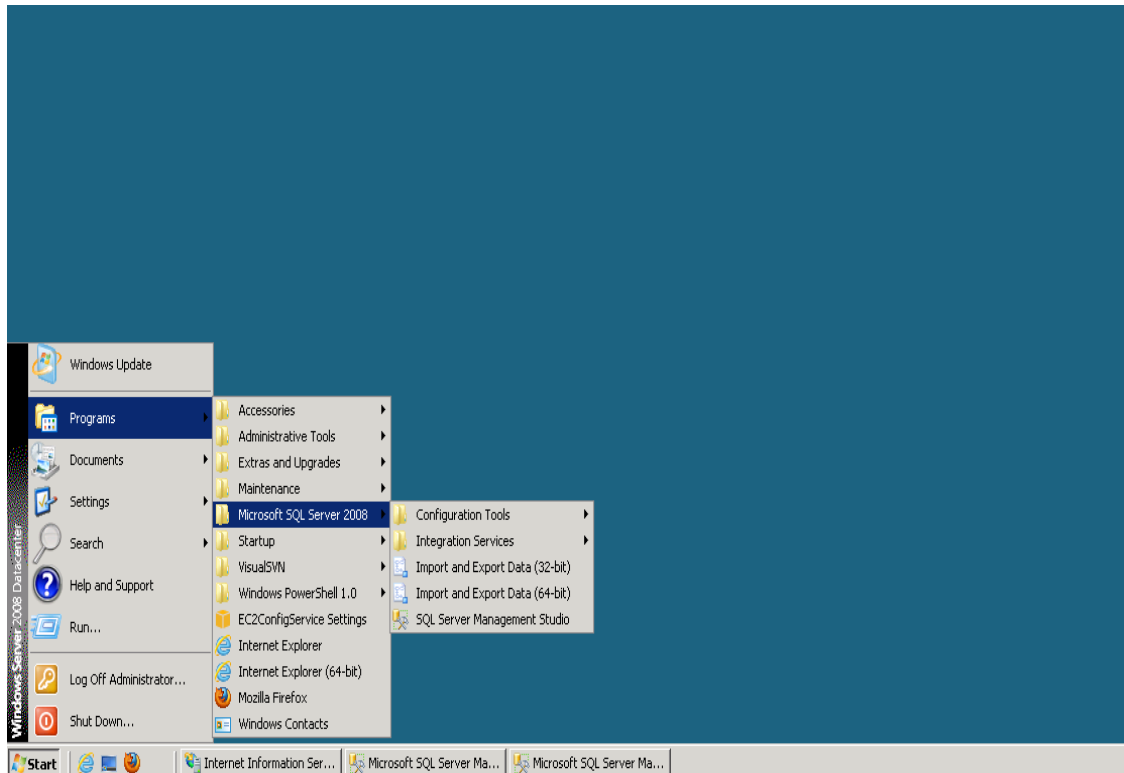
12. Now verify the backup file.
13. You have successfully created the SDAC backup file at the selected location. You will need this database backup file location if you want to revert back the changes of this release to a previous version.

Run the Database Script on SDAC database

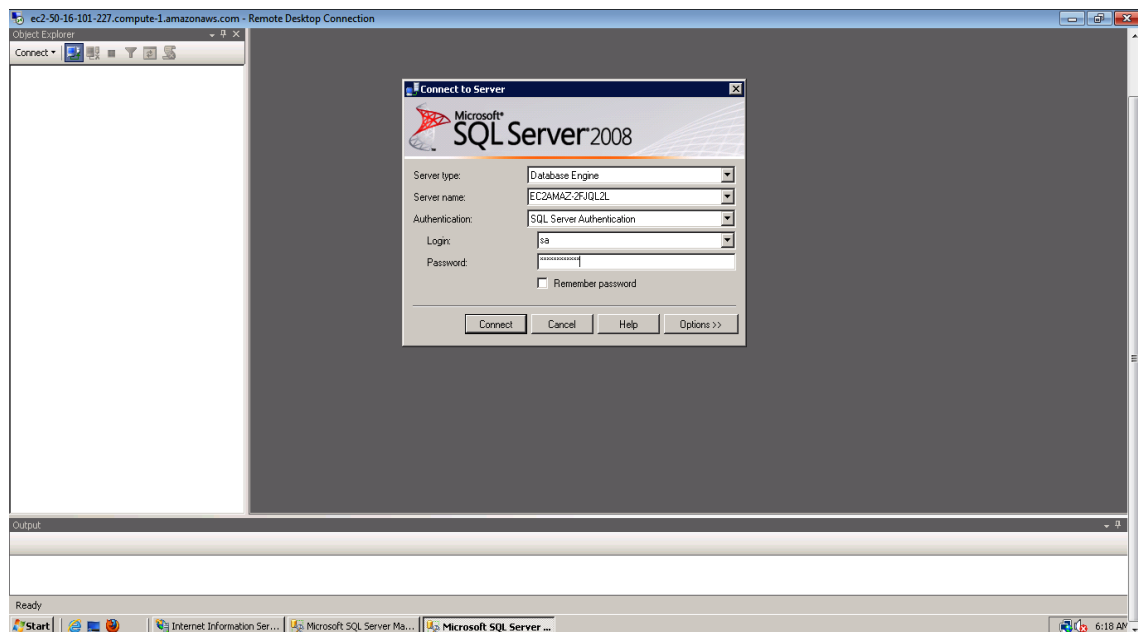
To run the database script provided with the release, please copy the SDAC_DB_Script folder from Database CD provided to local drive of the server (you can delete the DB Script folder after completion of this section). If you don't have remote desktop access to Database Server then you can connect to Database Server through SQL Server Management Studio from your workstation (if access provided)

1. Open the **SQL Server Management Studio** from **Start Menu → Programs → Microsoft SQL Server 2008 → SQL Server Management Studio**

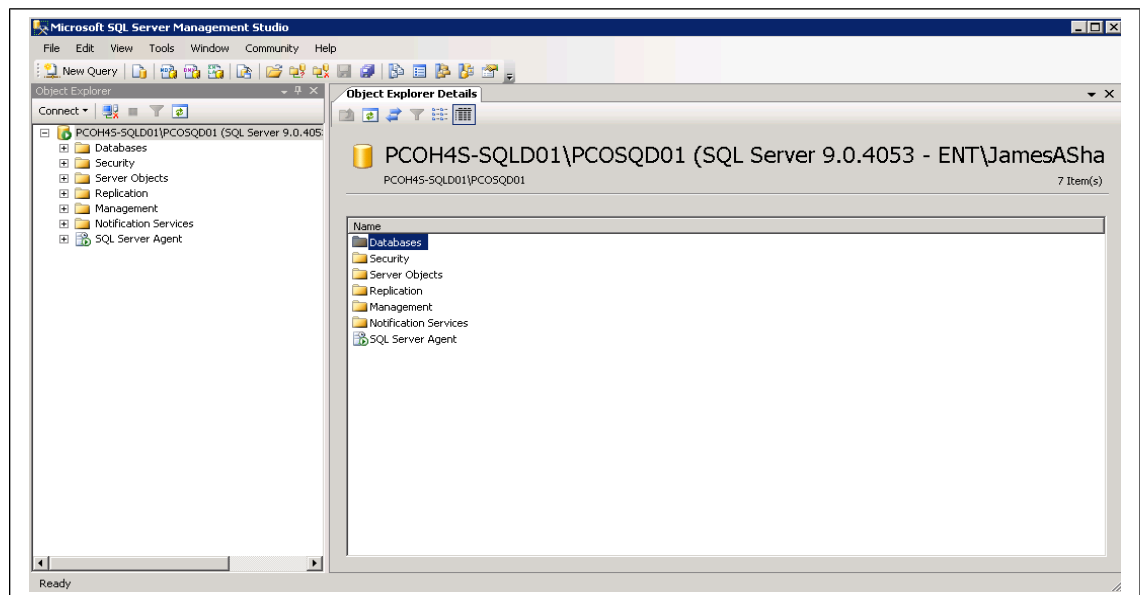
See the below figure.



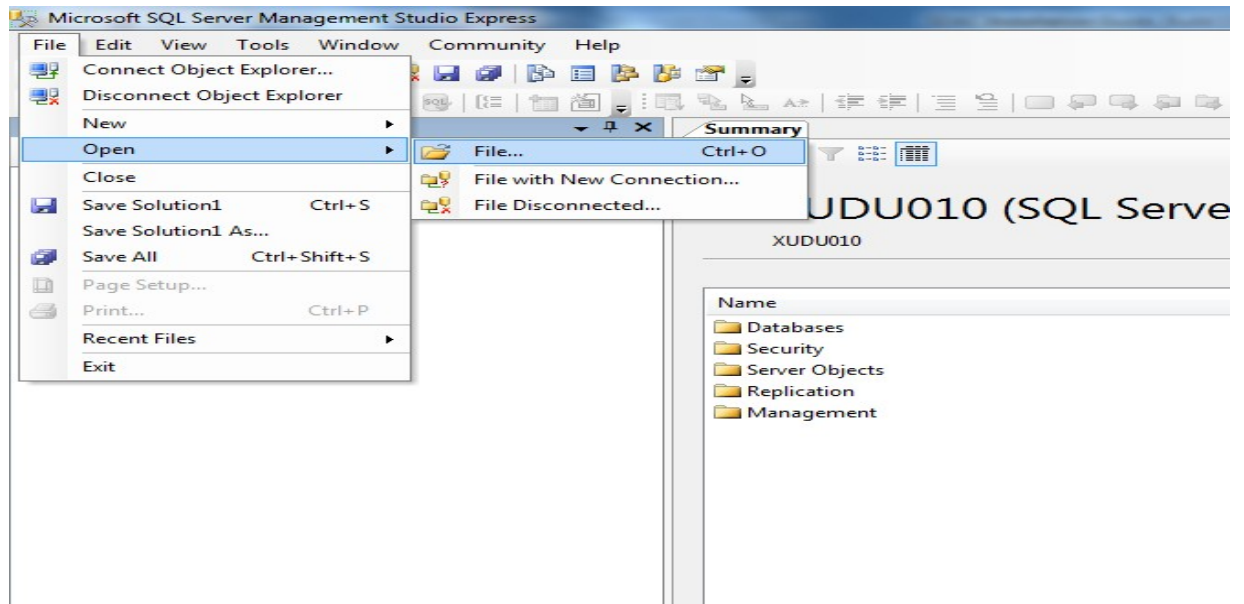
2. It will open the Microsoft SQL Server Management Studio window and will prompt for Connect to Server. Select the valid Server Name and Authentication method to login to SQL Server. Click on Connect button to login to SQL Server 2008. See the below figure.



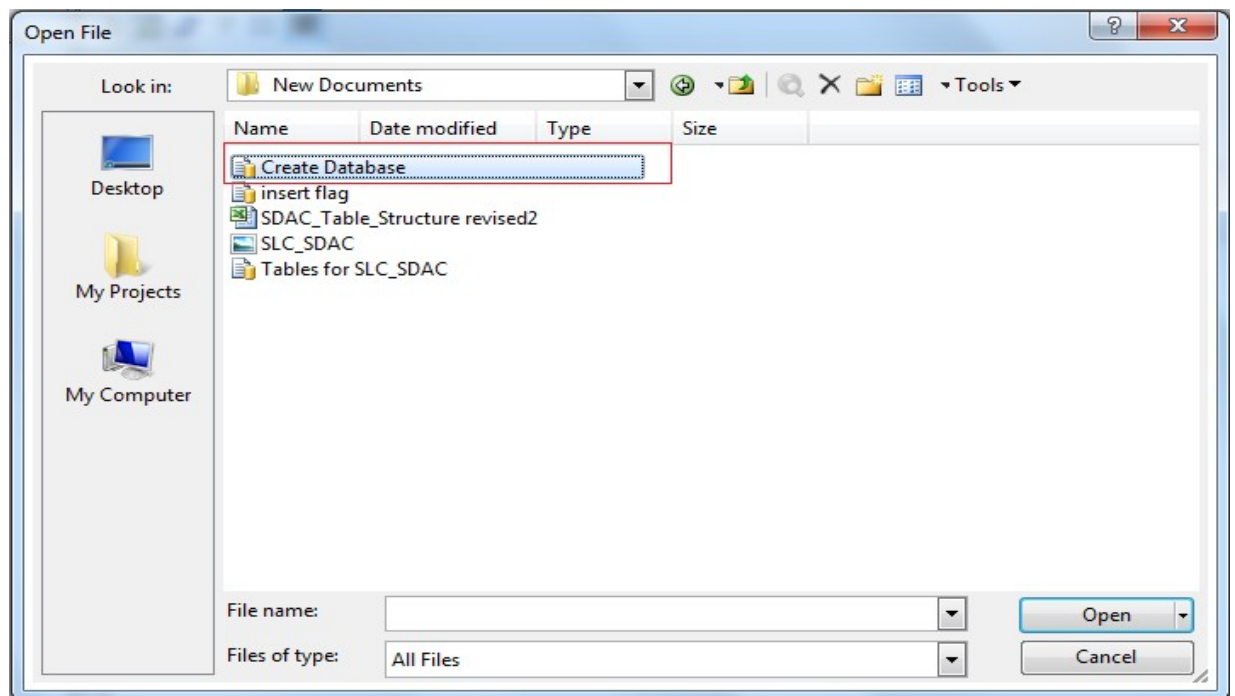
3. After clicking on Connect button you will see the below figure.



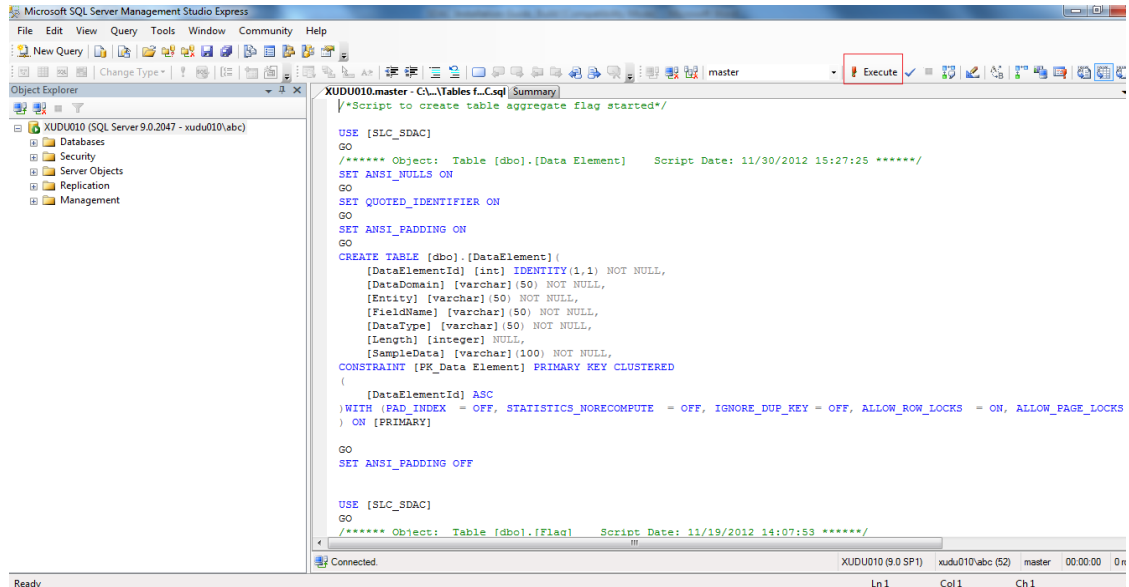
4. Open the database script [Create SDAC Database.sql](#) by selecting **File → Open → File...** or by pressing **Ctrl + O** shortcut keys.



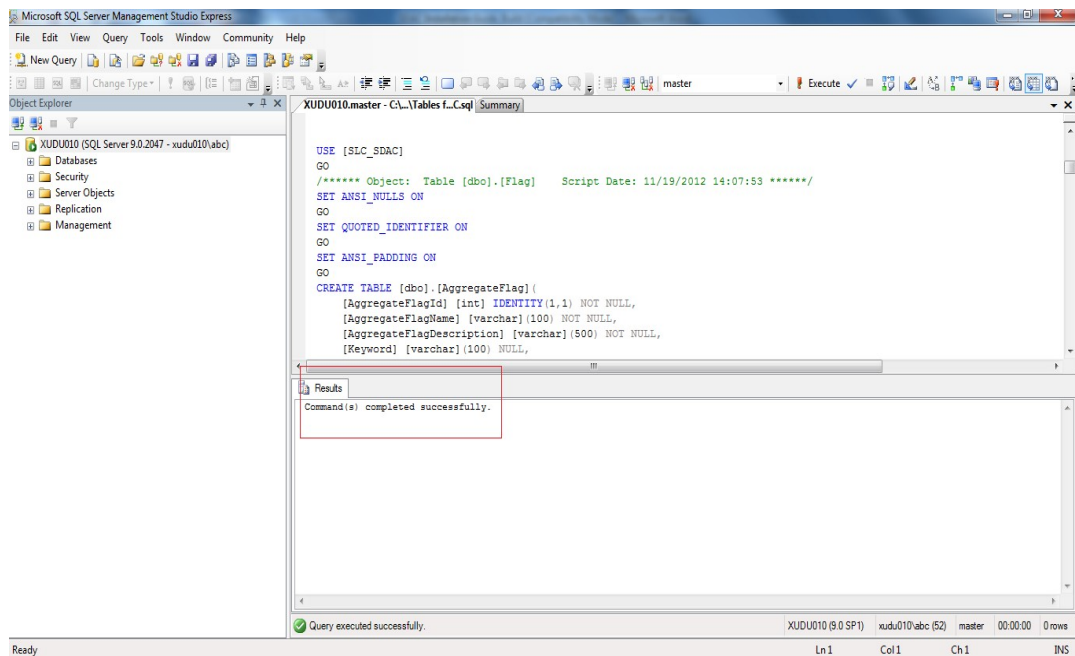
5. Please make sure you select the correct Database Script.
6. Select the database script in Open File window from the local drive where you copied the scripts from CD. After selecting the Database Script file name click on Open button.



7. This will open the Database script as shown in below figure. **Click on Execute** button to run the database script as shown in below figure.



8. Once execution of query is completed you will see the result of the database script at bottom of the page as shown in below figure.



9. Scroll down the message section as shown in the above screen and check for any error messages. (Please Ignore any warning message)
10. In the same manner run the other scripts.
11. You have successfully updated the SDAC database with the provided Database Script.
12. Close the SQL Server Management Studio.

NOTE: You can use above steps to run any Database Script using SQL Server Management Studio if any is provided along with release.

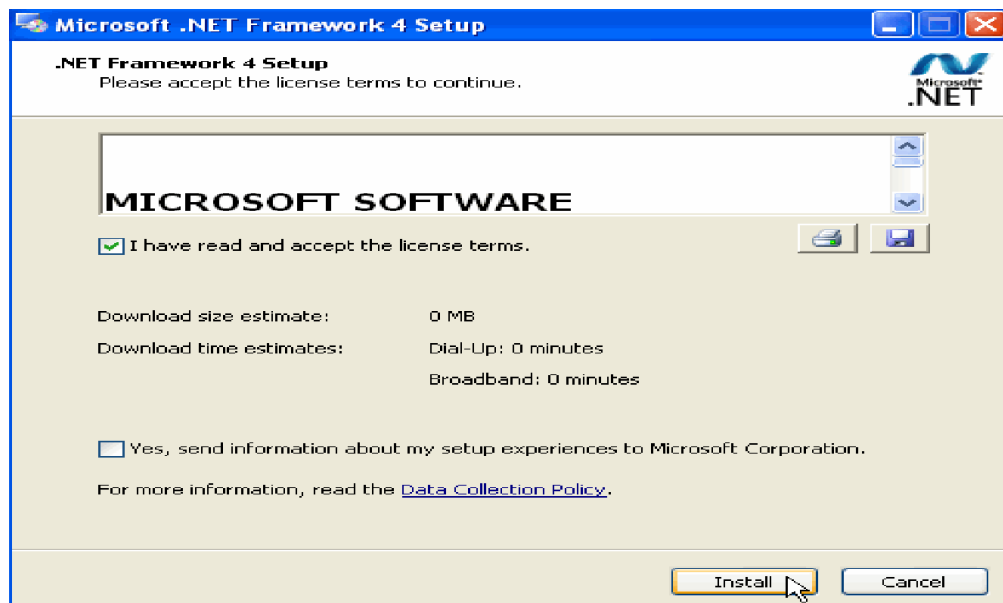
Internet Information Services (IIS)

Install Microsoft .NET Framework 4.0 Setup

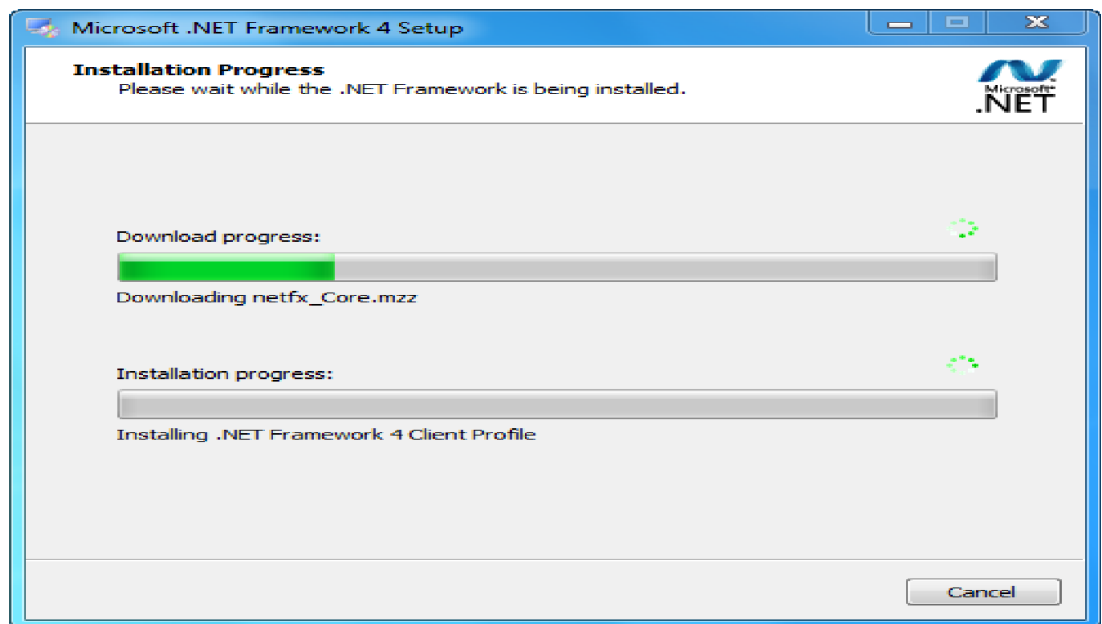
The Microsoft .NET Framework 4.0 is required in order to run SDAC application.

Please follow the steps below to install .NET Framework on the Application Server.

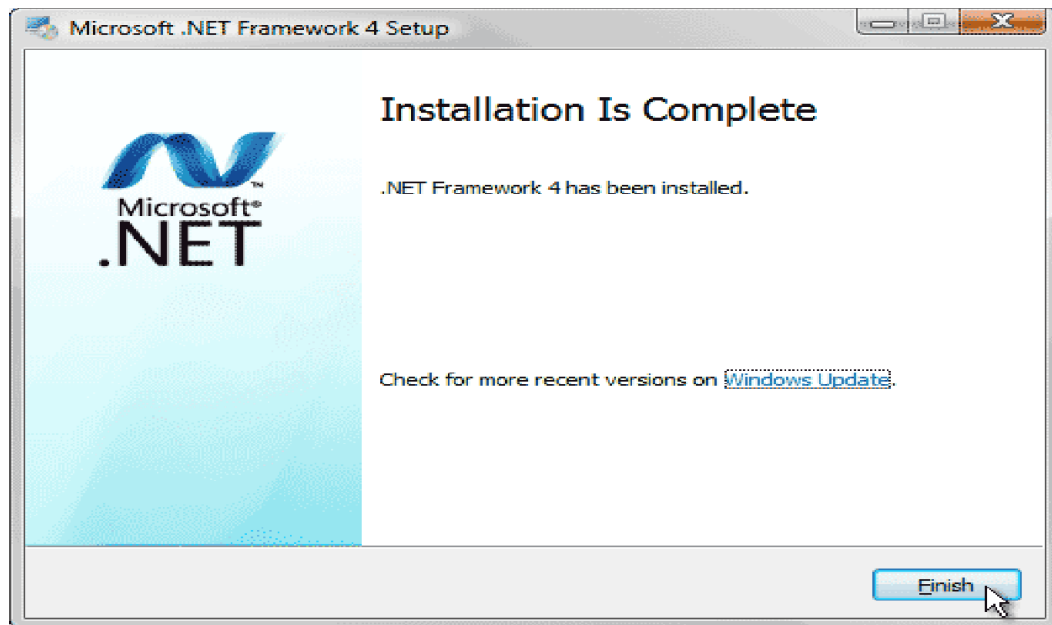
1. You can download **Microsoft .NET Framework 4.0 Service Pack 1** directly from the Microsoft website or copy the **“MS .NET Framework 4.0”** folder from Application Installation files provided to the local drive.
2. The **“MS .NET Framework 4.0”** folder contains the setup file
3. Double click on the **setup** file to start installation of .NET 4.0 Framework.
4. It will show the “Welcome to Setup” screen as shown in below figure.



5. Select **“I have read and ACCEPT the terms of the License Agreement”** and click on **Install** button.
6. It will show **Download and Install in Progress** screen as shown in below figure.



7. After installation, it will show **Setup Complete** screen as shown in below figure. Click on Exit button.



8. You have successfully installed the Microsoft .NET Framework 4.0 on Application Server.

Publish SDAC in IIS 7

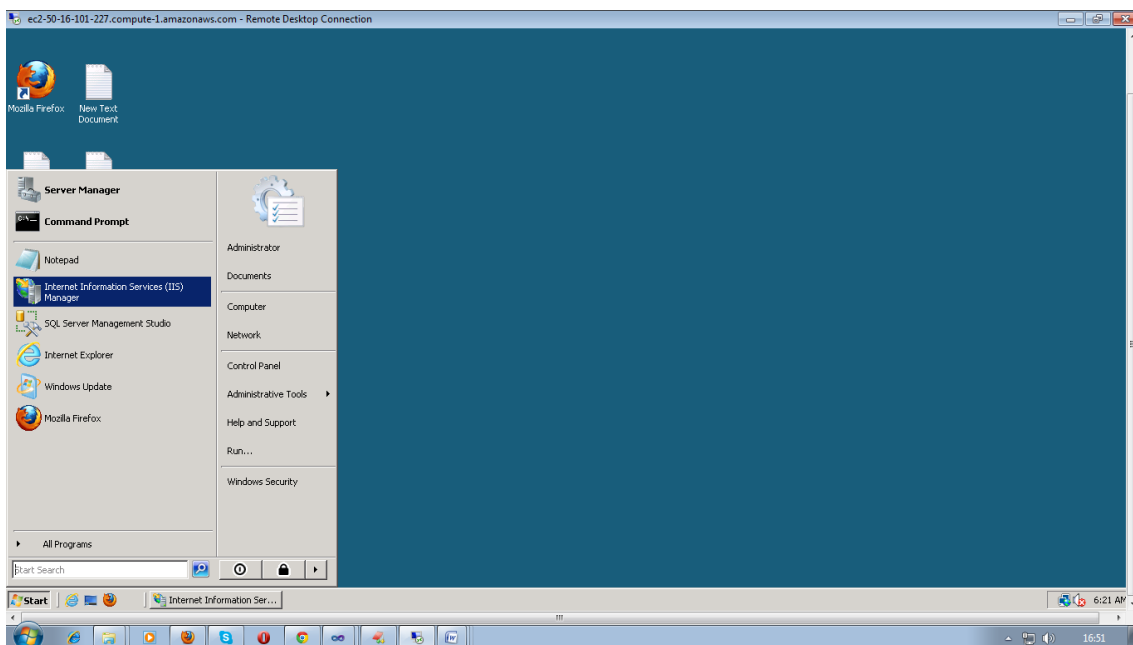
The following article explains how to create a new website on your Windows server in IIS 7. IIS 7 (Internet Information Services) controls the configuration of each website on the server. There are two types of websites that can be configured, IP based sites and Name based sites*.

Step 1: Log into your server through Remote Desktop Connection

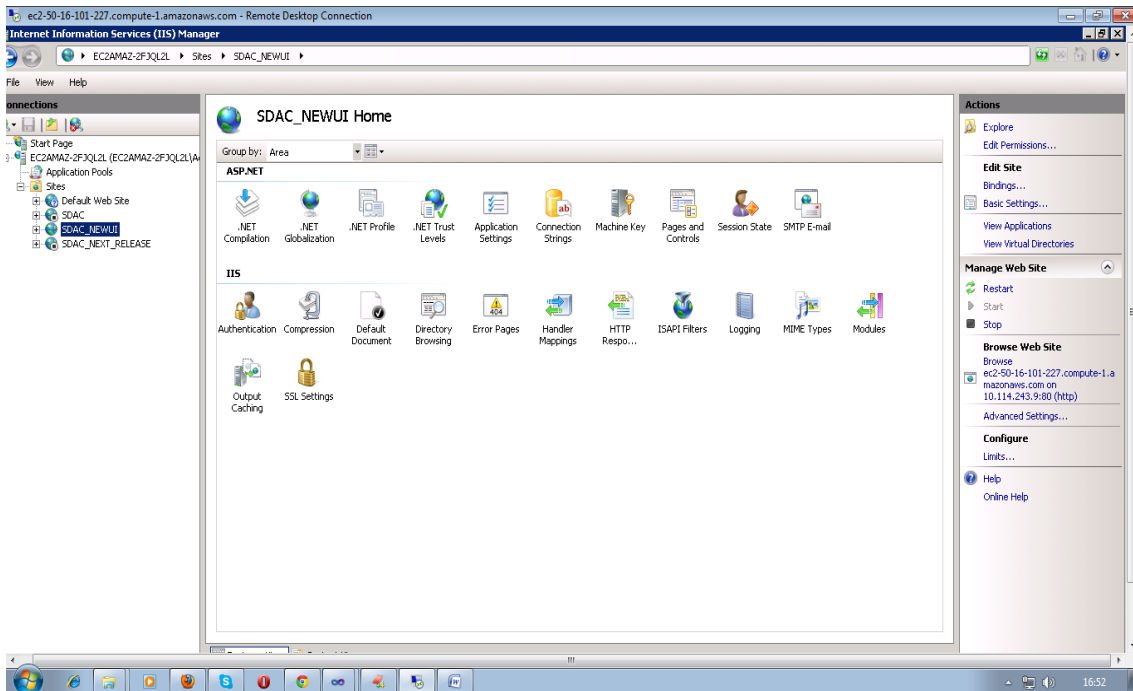
Step 2: From Visual Studio, publish your Web application.

Step 3: Copy the published application folder to "C:\inetpub\wwwroot" [default] folder.

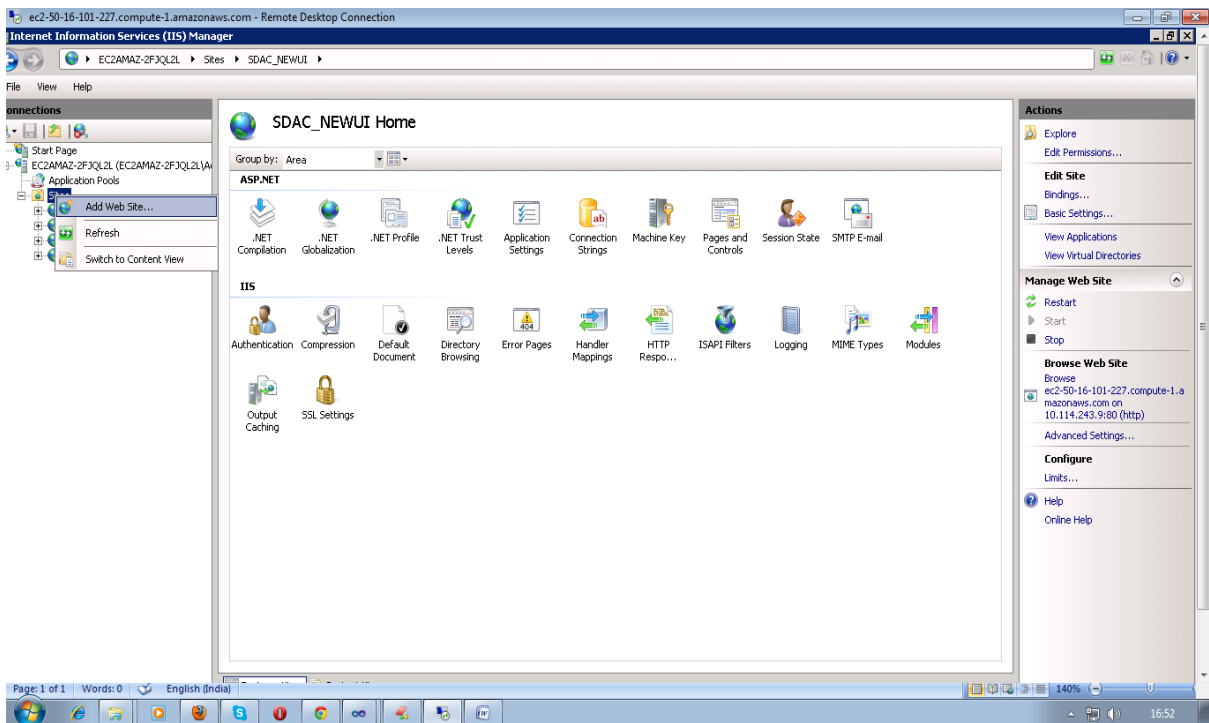
Step 4: From **RUN - > inetmgr -> OK** or **Use Start-> Internet Information Services (Manager)**



The following screen will come. This is the main page for any application, where you can add your new website or configure your website in IIS.



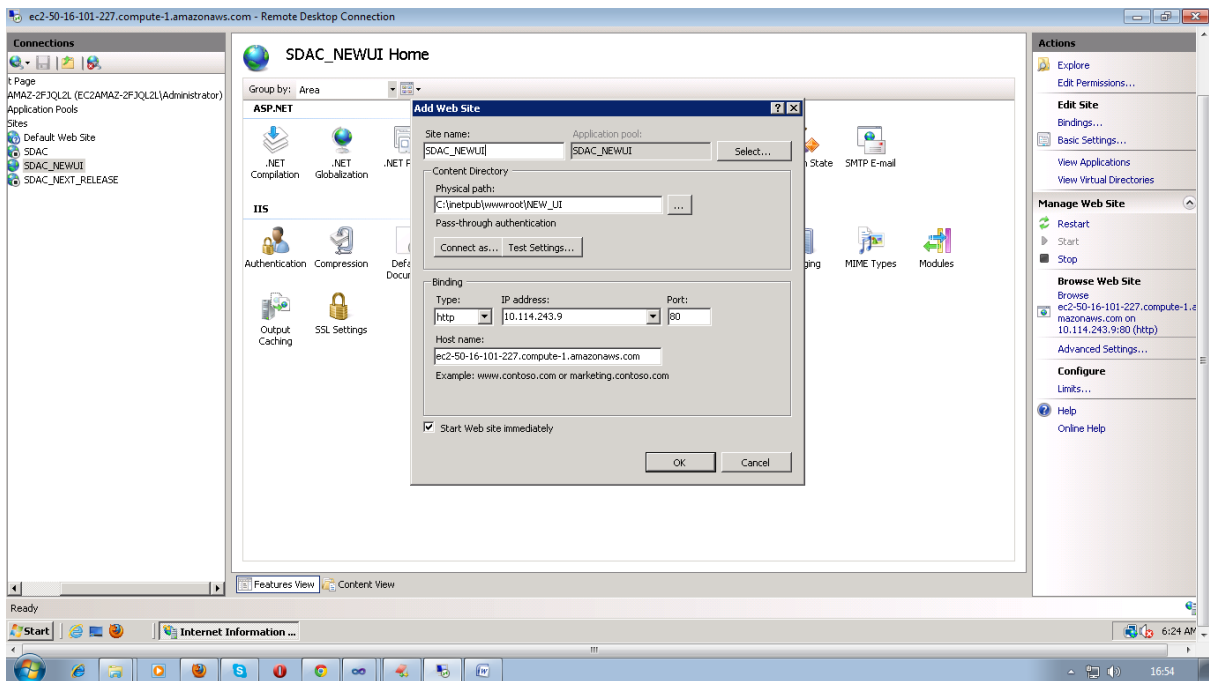
Step 5: To add new website Right click on Sites Choose Add Website option



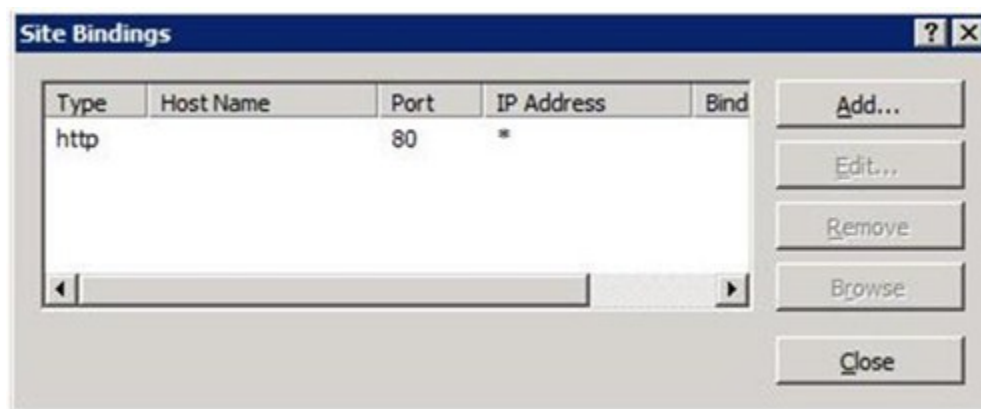
Step 6: Type in the Site Name. A description or the domain name is best to put into this field. This will automatically create an Application Pool for the site.

Step 7: Enter or browse to the path where the website will have its root directory. This is the directory where the home page should go.

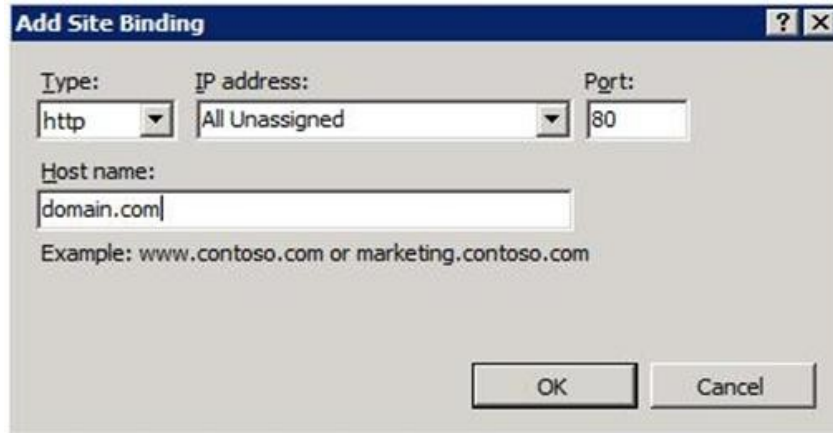
Step 8: For the Binding, choose HTTP, choose an IP address that is free, leave the port as port 80, and enter in your domain name (i.e. www.domain.com) as the Host Name. Leave the checkbox for Start Web site immediately checked and click OK.



Step 9: Go back to the IIS7 main screen and select the site under the Sites folder. Click Bindings... on the right sidebar so there is a host header for domain.com as well.



Step 10: Click the Add button, choose HTTP, use the same IP address as setup in step 7, leave the port as port 80, and enter in your domain name (i.e.domain.com) as the Host Name. Click OK



The screenshot shows a Windows-style dialog box titled "Add Site Binding". It has three dropdown menus at the top: "Type:" set to "http", "IP address:" set to "All Unassigned", and "Port:" set to "80". Below these is a text box for "Host name:" containing "domain.com". Underneath the text box is an example: "Example: www.contoso.com or marketing.contoso.com". At the bottom right are "OK" and "Cancel" buttons.

An IP-based website has a unique IP address therefore no other website can use this IP address. If you need multiple IP-based websites, you will need to contact us to add the IP addresses to your server.

A name-based website shares a single IP address with each website on the server. You are able to make as many name-based sites as you wish with your single IP address.

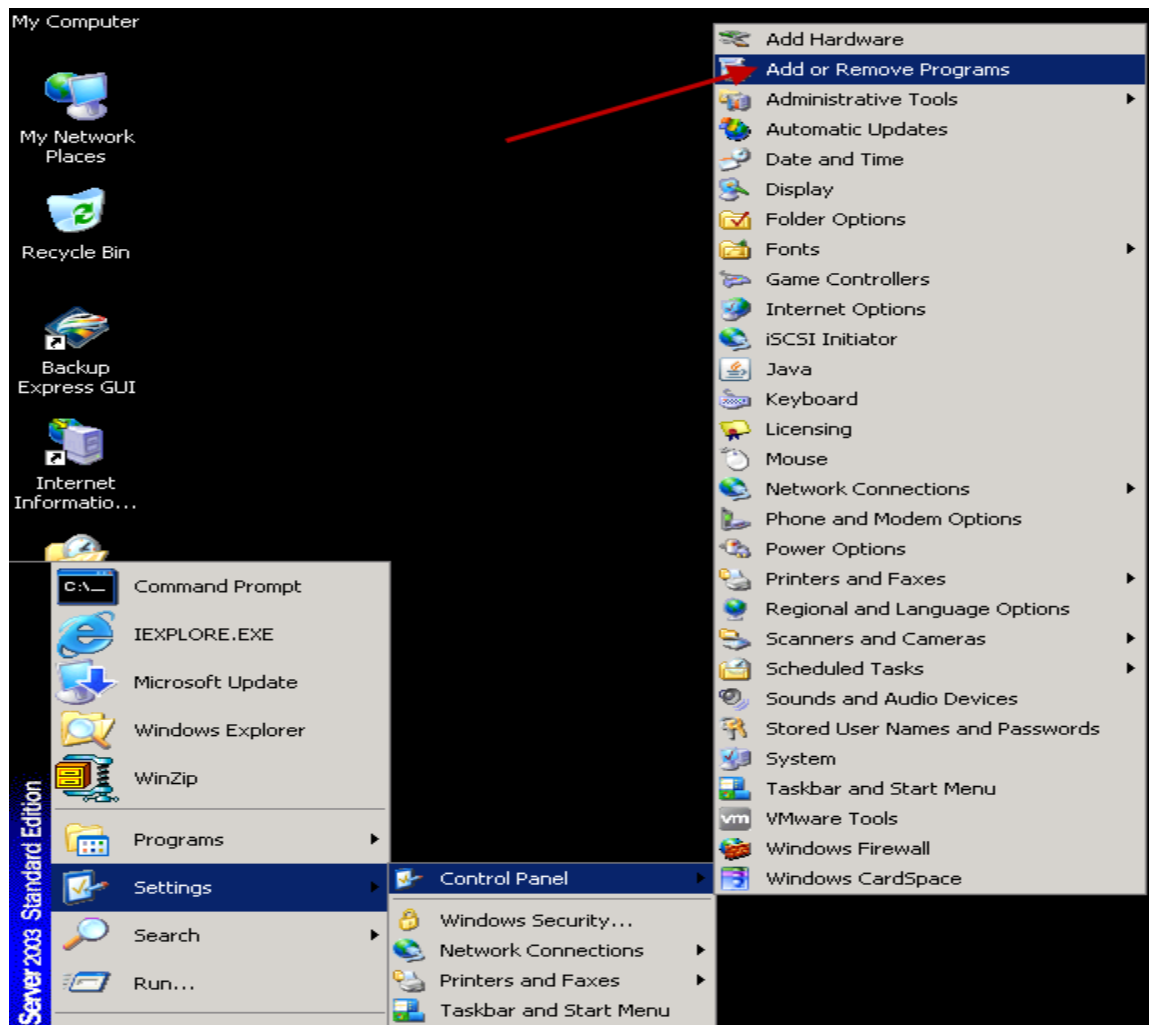
Un-Installation

Un-Install Microsoft .NET Framework 4.0

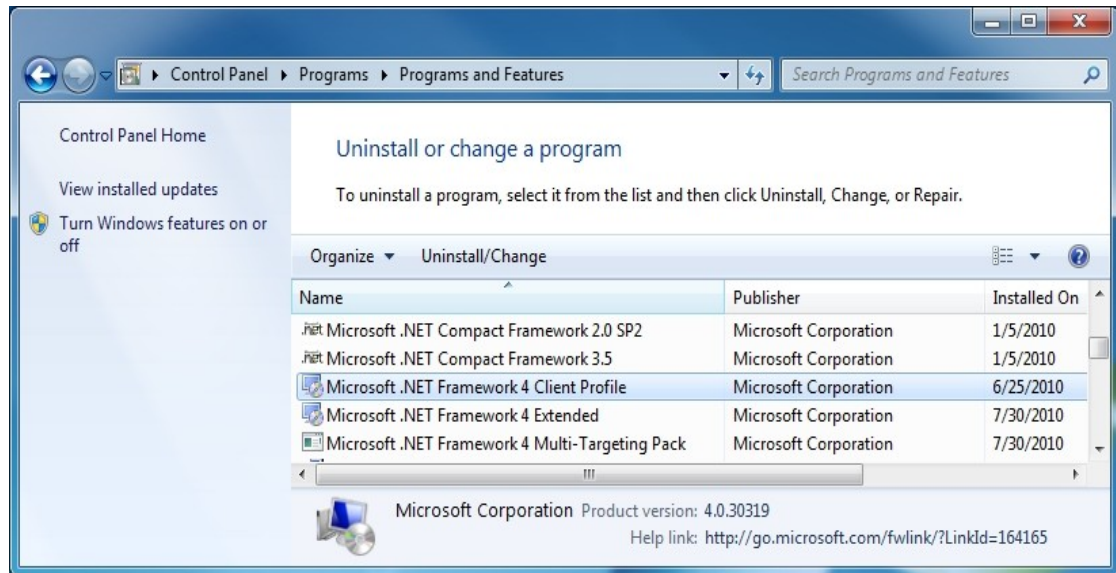
Follow the instructions in this section to uninstall the Microsoft .Net Framework 4.0 on Application Server.

If necessary, you can uninstall the Microsoft .NET Framework 4.0 using the following procedures. If you keep .NET framework 4.0 installed on the Application Server it will not impact anything on the server.

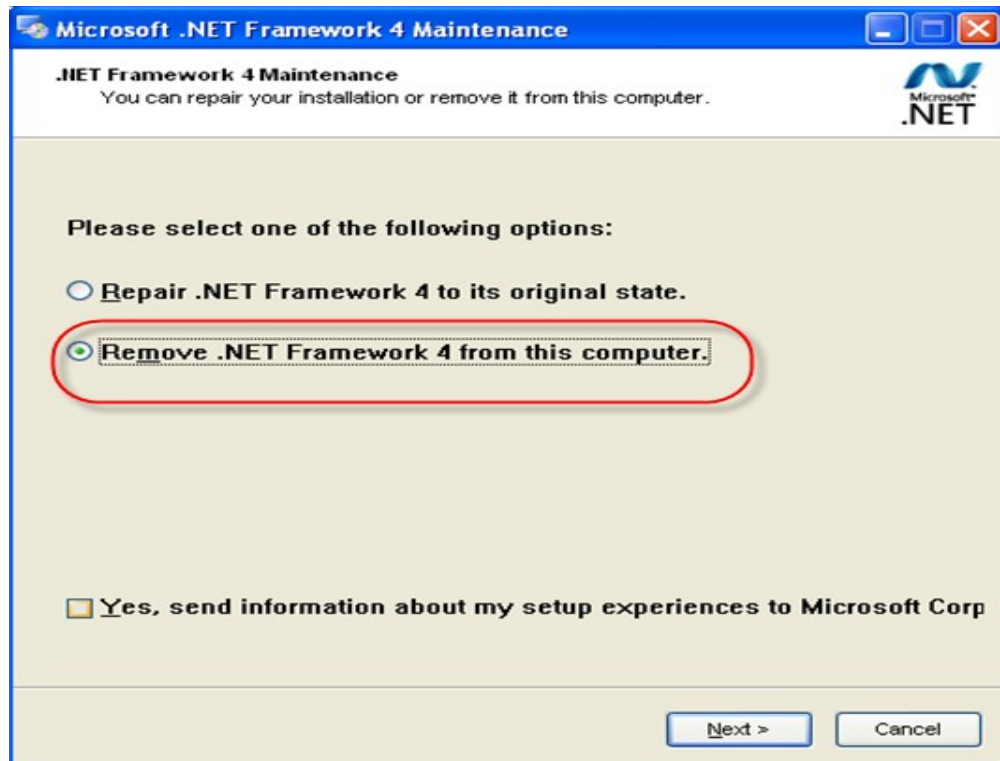
1. Open the **Add or Remove Programs** from **Start Menu → Settings → Control Panel → Add or Remove Programs** as shown in below figure.



2. It will open the **Add or Remove Programs** window and select **Microsoft .NET Framework 4.0** from the program list and click **Change/Remove** button as shown in the below figure.



3. It will show you Maintenance Mode window of Microsoft .NET Framework 4.0 Setup as shown in the below figure



4. Select **Uninstall** option and click on **Next** button to uninstall the framework from the server.
5. It will show you below confirmation message. **Click** on **Ok** button.
6. After clicking on **OK** button, it will start removing Microsoft .NET Framework 4.0 setup from the server.
7. After completion, it will show below figure. **Click** on **Finish** button.



8. You have successfully removed the Microsoft .NET Framework 4.0 setup from Application Server.

Register SDAC

Steps for Application Registration

1. Open the inBloom sandbox portal application by pointing to the URL:
<http://portal.sandbox.inBloom.org/>
2. Login to the sandbox using an Administrator-level user account



inBloom
sandbox

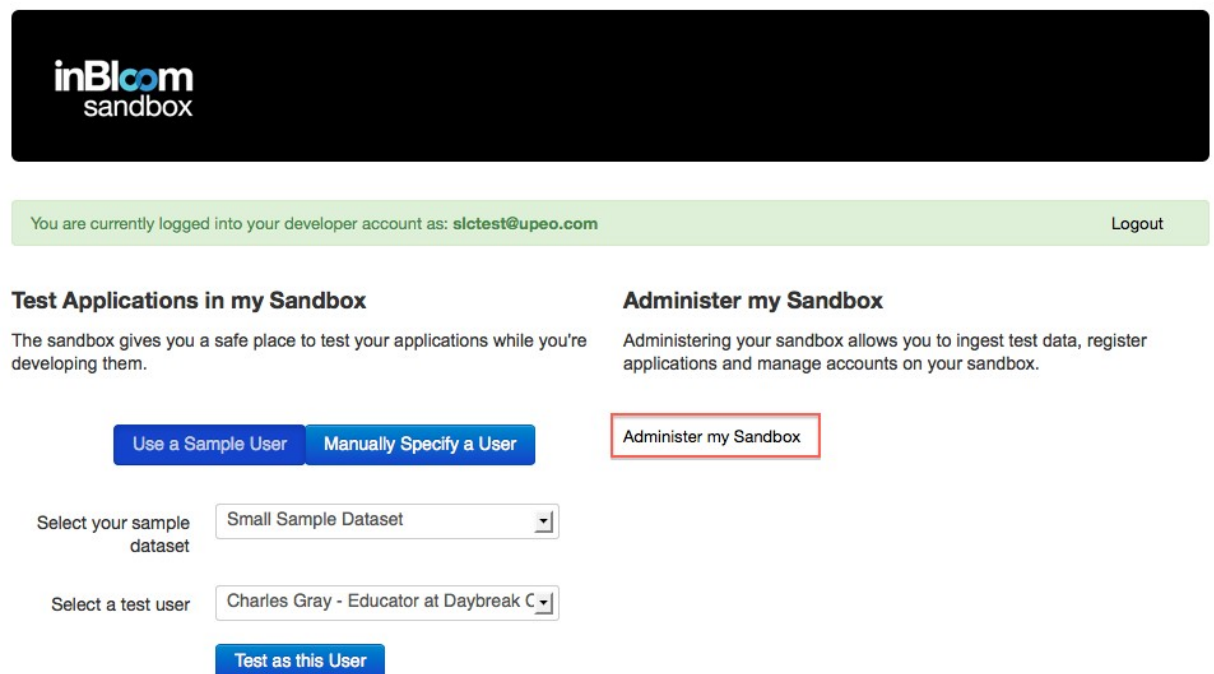
Email Address:

Password:

[Forgot your password?](#)

[Sign in](#)

3. After successful login click on Administer my Sandbox



inBloom
sandbox

You are currently logged into your developer account as: **slctest@upei.com** [Logout](#)

Test Applications in my Sandbox

The sandbox gives you a safe place to test your applications while you're developing them.

[Use a Sample User](#) [Manually Specify a User](#) [Administer my Sandbox](#)

Select your sample dataset:


Select a test user:

[Test as this User](#)

Administer my Sandbox

Administering your sandbox allows you to ingest test data, register applications and manage accounts on your sandbox.

4. Then click on the Admin menu option at the top

slctest@upeo.com ▾ **Admin**

HOME

Welcome

inBloom provides tools and applications to monitor, manage and boost student performance. As more applications become available, you will be better able to deliver top-notch instruction to kids through applications that provide valuable insights and also save you time.

The applications that you can use are listed on the right side of the page.

A Word About Student Privacy

inBloom contains personal data about students. This is sensitive stuff, and it's your role to keep this data safe. With that, here are some things to keep in mind:

- Only share student data with people who are authorized by federal and/or local law.
- Do not share your password.
- Update your password periodically.
- Log out each time you're done working. Click your name at the top of the page to logout.

We're Listening

If you run into any issues or you have a good idea for a new feature, please contact us. Just click your name in the upper right, then click [Report a problem](#)


My Applications

inBloom Resource Tagger

The inBloom Resource Tagger allows users to describe online Learning Resources and submit these Resource Descriptions to the inBloom Learning Registry Index, where they will then be accessible via the inBloom Resource Search application.

© 2012 inBloom, Inc. and its affiliates [Privacy Policy](#) [Terms of Use](#)

5. Go to the application management by clicking on Application Registration

slctest@upeo.com ▾ | Admin

ADMIN

Welcome

Welcome to inBloom Admin Suite. Here, you can configure systems settings. Also, depending on your role, you may be able to access configuration options for some approved applications. Not all applications allow configuration access here. so do not be alarmed if you see all application below.

System Tools

- **Register Application**
- Create Custom Roles
- Change Password

Application Configuration

© 2012 inBloom, Inc. and its affiliates [Privacy Policy](#) [Terms of Use](#)

6. Register a new application by clicking on new application button

inBloomsicrest@upeo.com ▾ | Admin

Manage Applications

Register a new application and update applications you have already registered.

New Application

Name	Version	Vendor	Creation Date	Last Update	
Student Data Aggregation Calculator	3.0	Upeo	Tue, 22 Jan 2013 08:50:24 +0000	Thu, 21 Feb 2013 15:44:41 +0000	<div>Edit Delete</div>

7. If you have already registered an application then update the application by simply clicking on the Edit button.

inBloomsicrest@upeo.com ▾ | Admin

Manage Applications

Register a new application and update applications you have already registered.

New Application

Name	Version	Vendor	Creation Date	Last Update	
Student Data Aggregation Calculator	3.0	Upeo	Tue, 22 Jan 2013 08:50:24 +0000	Thu, 21 Feb 2013 15:44:41 +0000	<div>Edit Delete</div>

8. Getting Client ID and Shared Secret
9. After entering application details, admin URL and other related information the Client ID and Shared Secret will automatically generate.

Edit Application

Change your application details

Client ID
The client ID needed to perform OAuth

eJ28fp8oDW

Shared Secret
The shared secret needed to perform OAuth

IP1f1huQLx6qIIXM7PVeTTObvfgpnAMuacOuUpX3U9PqRGBm

Name

SDAC

Description

Student Data Aggregation
Calculator

10. Redirect URL - Where OAuth will redirect to after authentication.

Version

1.0

The current version of your application (1.0)

Url

http://localhost/Search.aspx

The URL used to access your application

Admin Url

https://

Where you can go to administer your application (Optional)

Admin Application

☐ Is this an administrative application?

Display Method

Full Window App

How does this application behave, can it be put into an IFrame, behave as a portlet, or stand alone?

Redirect Url

http://localhost/Search.aspx

Where OAuth will redirect to after authentication.

Installed

☐ This is used for applications that aren't web-based (Mobile).

Image Url

An optional image url to be displayed as an icon

The Web.config file

Web.config file, is a configuration file for the ASP.NET web application. An ASP.NET application has one Web.config file which keeps the configurations required for the corresponding application. Web.config file is written in XML with specific tags having specific meanings.

What can be stored in Web.config?

There are a number of important settings that can be stored in the configuration file. Here are some of the most frequently used configurations, stored conveniently inside Web.config.

1. Application Setting
2. Database Connection

Application Setting

This section is used to store custom application configuration like database connection strings, file paths, etc. This also can be used for custom application-wide constants to store information over multiple pages. It is based on the requirements of the application.

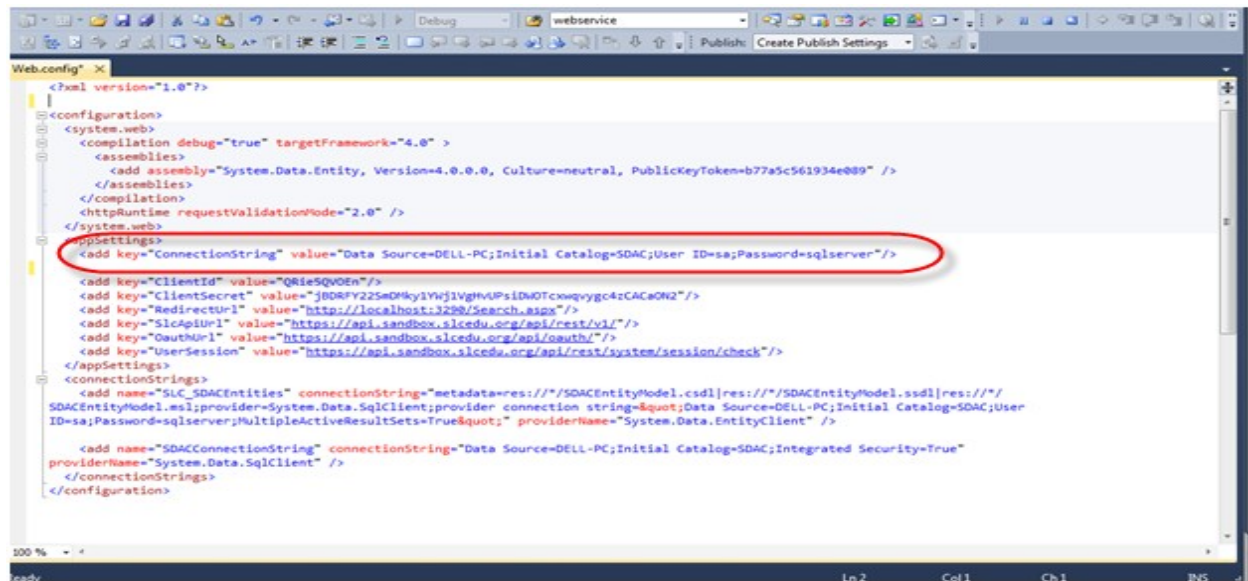
Connection String

When your application connects to a database or a data file you let ADO or ADO.NET utilize a provider to do the job for you. The connection string contains the information that the provider needs to know to be able to establish a connection to the database or the data file.

Because there are different providers and each provider has multiple ways to make a connection there are many different ways to write a connection string. It's like the address when sending a regular mail. Depending on the origin and destination and who is going to make the transport you need to write down the address in different ways.

In order to set up a connection string, you need to set the values for the `ConnectionString` key first. See the screen shot below.



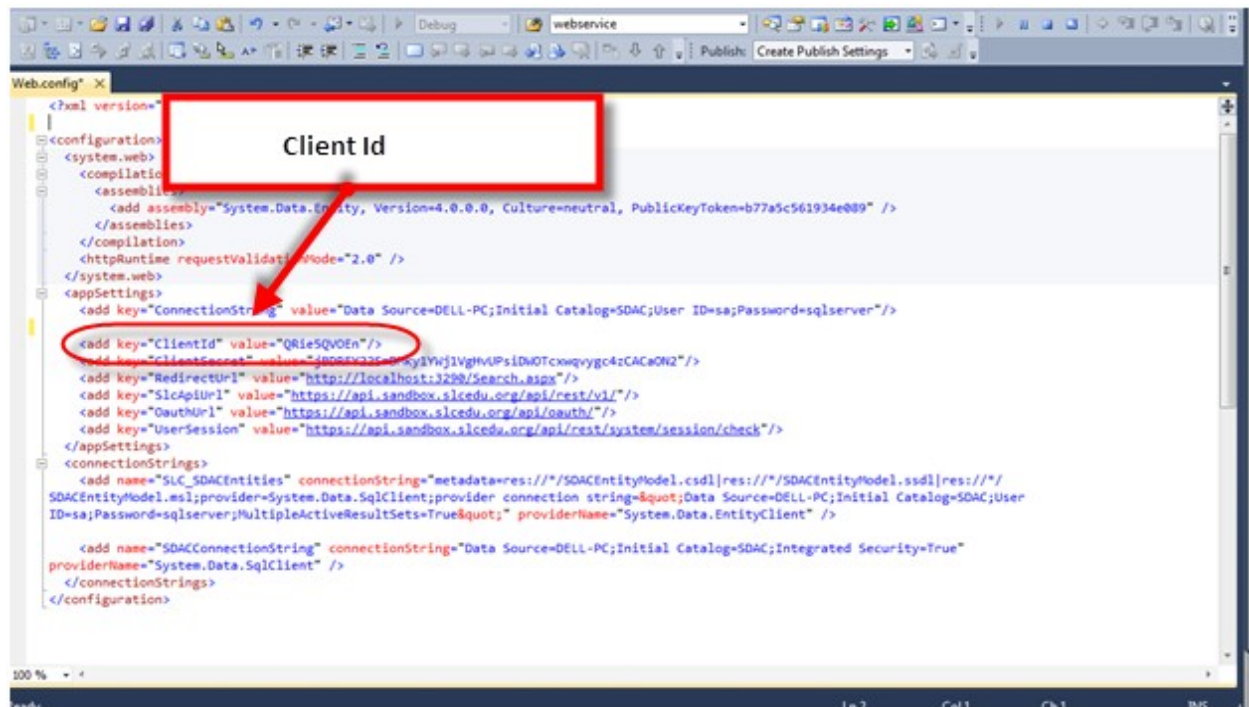


```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" >
      <assemblies>
        <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e009" />
      </assemblies>
    </compilation>
    <httpRuntime requestValidationMode="2.0" />
  </system.web>
  <appSettings>
    <add key="ConnectionString" value="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver"/>
    <add key="ClientId" value="QRieSQVOEn"/>
    <add key="ClientSecret" value="j8DRFv225MD%ky1VnJ1VghN4UPsIDuOTcxwqvygc4zCACwON2"/>
    <add key="RedirectUrl" value="https://localhost:3259/Seach.aspx"/>
    <add key="SicApiUrl" value="https://api.sandbox.slcedu.org/api/rest/v1/" />
    <add key="OAuthUrl" value="https://api.sandbox.slcedu.org/api/oauth/" />
    <add key="UserSession" value="https://api.sandbox.slcedu.org/api/rest/system/session/check"/>
  </appSettings>
  <connectionStrings>
    <add name="SLC_SDACEntities" connectionString="metadata=res://*/SDACEntityModel.csdl|res://*/SDACEntityModel.ssdl|res://*/SDACEntityModel.ssi|provider=System.Data.SqlClient;provider connection string="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver;MultipleActiveResultSets=True" providerName=System.Data.EntityClient" />
    <add name="SDACConnectionString" connectionString="Data Source=DELL-PC;Initial Catalog=SDAC;Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

Client ID

One of the new features being added to version 4.1 of ASP.NET is the ability to control the client side IDs that are generated by the framework. Previously the framework would modify the client side IDs to uniquely identify each control. This sometimes left you with the ID you defined in markup or sometimes left you with something that looks like this, "ctl00_MasterPageBody_ctl01_Textbox1."

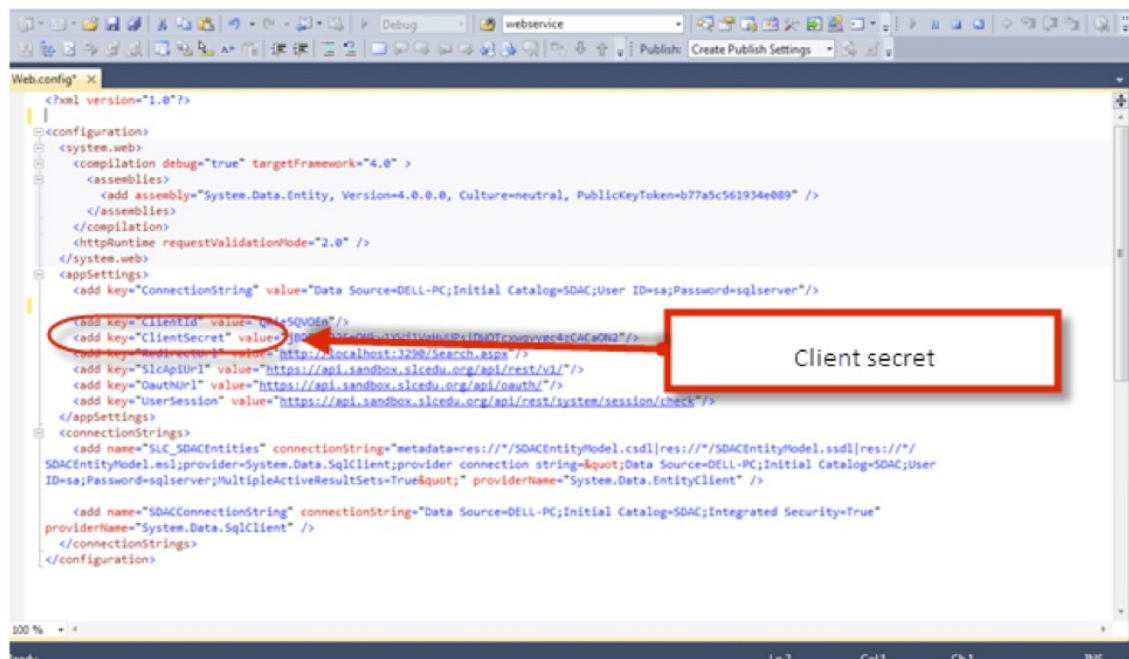
The modification of the client side ID property works great to ensure that each element is uniquely identified, however, to anyone that has tried to do any sort of client side scripting this becomes very frustrating. Chances are that if you have worked in ASP.NET for any time at all you have run into this issue. The problem is that until runtime you do not what the client side ID could be, making it difficult to do any kind of client side scripting. In addition any modification of the page, adding removing controls, can result in a different client side ID being generated.



The screenshot shows a Visual Studio IDE with a web service project. The 'Web.config' file is open, displaying XML configuration. A red box highlights the 'Client Id' value in the 'appSettings' section, with a red arrow pointing to it from the text 'Client Id' above. The 'Client Id' value is 'QRieSQVOEn'.

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation>
      <assembly name="System.Data.Entity, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
    </compilation>
    <httpRuntime requestValidationMode="2.0" />
  </system.web>
  <appSettings>
    <add key="ConnectionString" value="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver"/>
    <add key="ClientId" value="QRieSQVOEn"/>
    <add key="ClientSecret" value="30C..."/>
    <add key="RedirectUrl" value="http://localhost:3290/Search.aspx"/>
    <add key="SicApiUrl" value="https://api.sandbox.slcedu.org/api/rest/v1"/>
    <add key="OauthUrl" value="https://api.sandbox.slcedu.org/api/oauth"/>
    <add key="UserSession" value="https://api.sandbox.slcedu.org/api/rest/system/session/check"/>
  </appSettings>
  <connectionStrings>
    <add name="SLC_SDACEntities" connectionString="metadata=res://*/SDACEntityModel.csdl|res://*/SDACEntityModel.ssdl|res://*/SDACEntityModel.msl;provider=System.Data.SqlClient;provider connection string="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver;MultipleActiveResultSets=True"; providerName=System.Data.EntityClient" />
    <add name="SDACConnectionString" connectionString="Data Source=DELL-PC;Initial Catalog=SDAC;Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

Client Secret ID After registering the application Client ID and Shared Secret ID are generated.

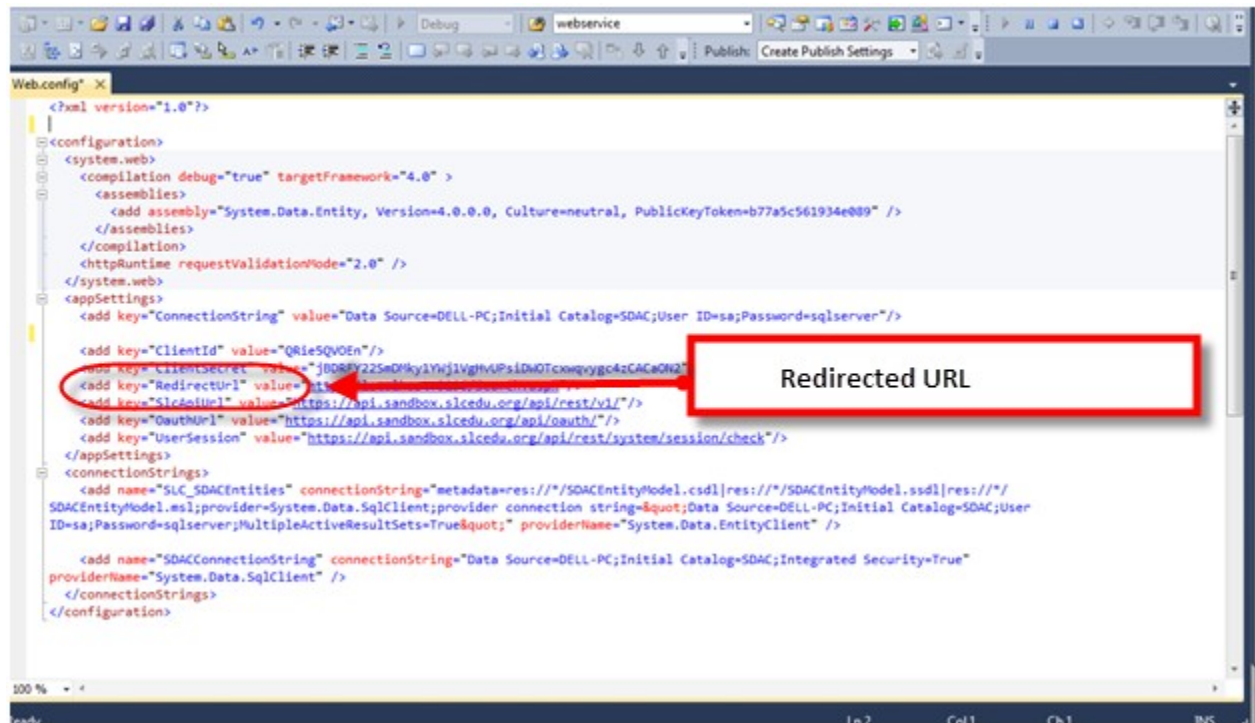


The screenshot shows the same 'Web.config' file as the previous image. A red box highlights the 'Client Secret' value in the 'appSettings' section, with a red arrow pointing to it from the text 'Client secret' to the right. The 'Client Secret' value is '30C...'. The 'RedirectUrl' is also visible as 'http://localhost:3290/Search.aspx'.

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" />
    <assemblies>
      <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
    </assemblies>
    <httpRuntime requestValidationMode="2.0" />
  </system.web>
  <appSettings>
    <add key="ConnectionString" value="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver"/>
    <add key="ClientId" value="QRieSQVOEn"/>
    <add key="ClientSecret" value="30C..."/>
    <add key="RedirectUrl" value="http://localhost:3290/Search.aspx"/>
    <add key="SicApiUrl" value="https://api.sandbox.slcedu.org/api/rest/v1"/>
    <add key="OauthUrl" value="https://api.sandbox.slcedu.org/api/oauth"/>
    <add key="UserSession" value="https://api.sandbox.slcedu.org/api/rest/system/session/check"/>
  </appSettings>
  <connectionStrings>
    <add name="SLC_SDACEntities" connectionString="metadata=res://*/SDACEntityModel.csdl|res://*/SDACEntityModel.ssdl|res://*/SDACEntityModel.msl;provider=System.Data.SqlClient;provider connection string="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver;MultipleActiveResultSets=True"; providerName=System.Data.EntityClient" />
    <add name="SDACConnectionString" connectionString="Data Source=DELL-PC;Initial Catalog=SDAC;Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

Redirected URL Is a URL where OAuth will redirect to after authentication.

Use rules to redirect domain aliases to one specific domain, or to specific folders (Windows Server)



```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" >
      <assemblies>
        <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
      </assemblies>
    </compilation>
    <httpRuntime requestValidationMode="2.0" />
  </system.web>
  <appSettings>
    <add key="ConnectionString" value="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver"/>
    <add key="ClientId" value="QRie5QVOEn"/>
    <add key="ClientSecret" value="jBDRFY225mDMky1YwJ1VgHvUPsiDWOTcxwqvygc4zCACA0N2"/>
    <add key="RedirectUrl" value="http://localhost:3290/Search.aspx"/>
    <add key="SlcApiUrl" value="https://api.sandbox.slcedu.org/api/rest/v1/" />
    <add key="OAuthUrl" value="https://api.sandbox.slcedu.org/api/oauth/" />
    <add key="UserSession" value="https://api.sandbox.slcedu.org/api/rest/system/session/check/" />
  </appSettings>
  <connectionStrings>
    <add name="SLC_SDACEntities" connectionString="metadata=res://*/SDACEntityModel.csdl|res://*/SDACEntityModel.ssdl|res://*/SDACEntityModel.msl;provider=System.Data.SqlClient;provider connection string="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver;MultipleActiveResultSets=True"; providerName="System.Data.EntityClient" />
    <add name="SDACConnectionString" connectionString="Data Source=DELL-PC;Initial Catalog=SDAC;Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

Entity Connection String

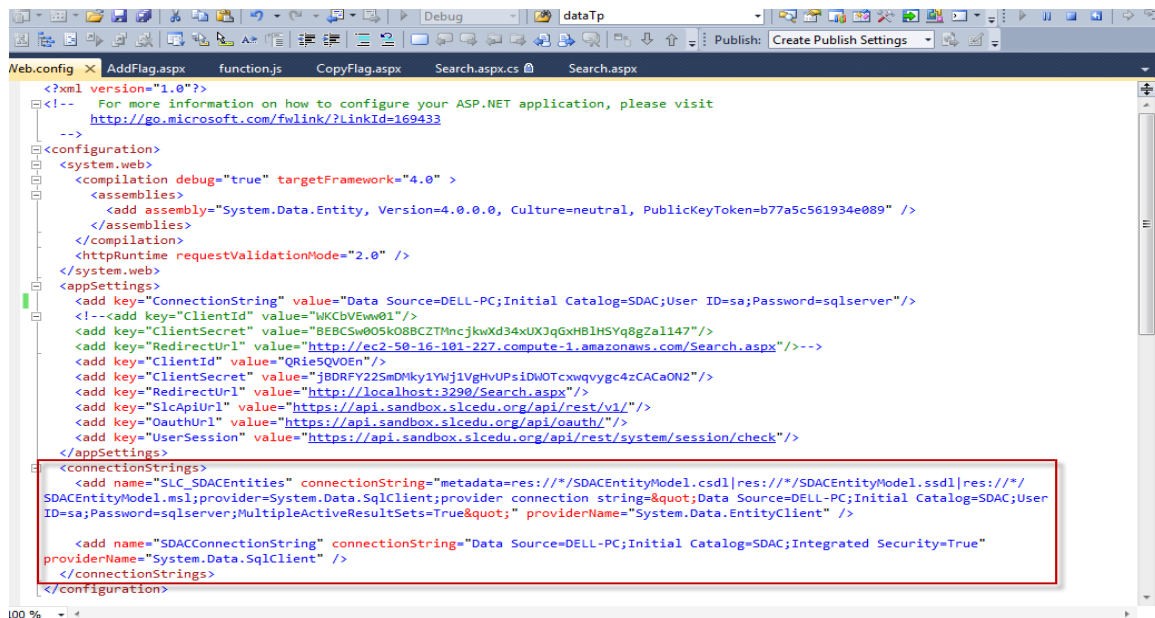
Used to connect the database to entity framework. It is created when creating an entity model and this string is pasted into the Web.config file.



```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" >
      <assemblies>
        <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
      </assemblies>
    </compilation>
    <httpRuntime requestValidationMode="2.0" />
  </system.web>
  <appSettings>
    <add key="ConnectionString" value="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver"/>
    <add key="ClientId" value="QRie5QVOEn"/>
    <add key="ClientSecret" value="jBDRFY225mDMky1YwJ1VgHvUPsiDWOTcxwqvygc4zCACA0N2"/>
    <add key="RedirectUrl" value="http://localhost:3290/Search.aspx"/>
    <add key="SlcApiUrl" value="https://api.sandbox.slcedu.org/api/rest/v1/" />
    <add key="OAuthUrl" value="https://api.sandbox.slcedu.org/api/oauth/" />
    <add key="UserSession" value="https://api.sandbox.slcedu.org/api/rest/system/session/check/" />
  </appSettings>
  <connectionStrings>
    <add name="SLC_SDACEntities" connectionString="metadata=res://*/SDACEntityModel.csdl|res://*/SDACEntityModel.ssdl|res://*/SDACEntityModel.msl;provider=System.Data.SqlClient;provider connection string="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver;MultipleActiveResultSets=True"; providerName="System.Data.EntityClient" />
    <add name="SDACConnectionString" connectionString="Data Source=DELL-PC;Initial Catalog=SDAC;Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

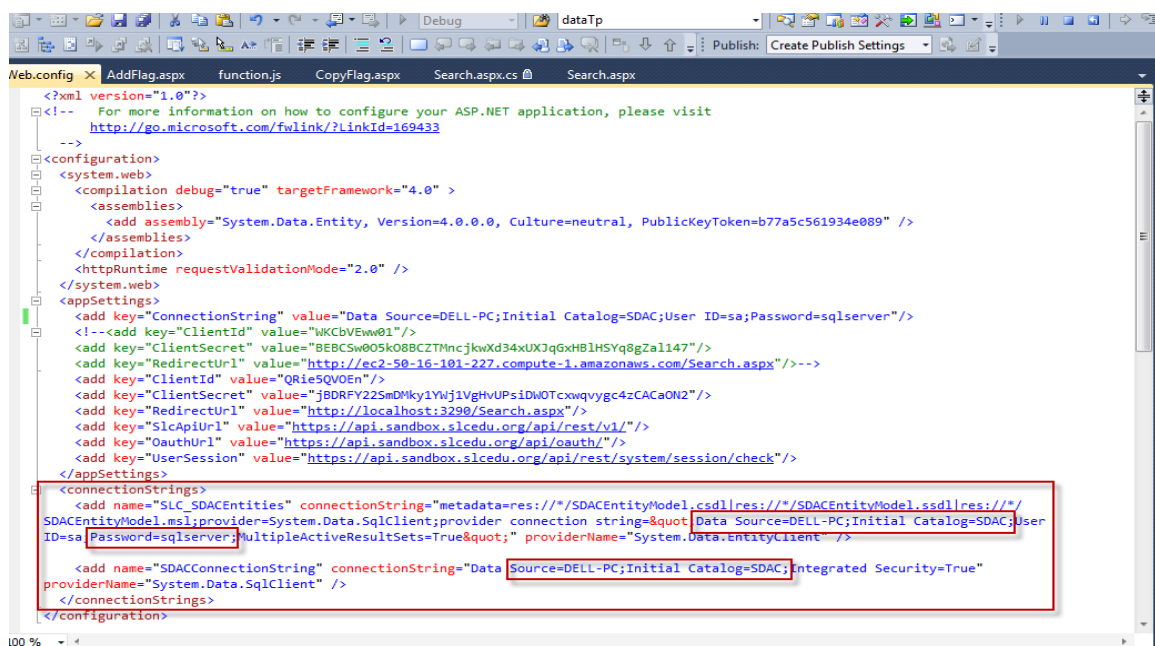
Database Connections

The most important configuration data that can be stored inside the Web.config file is the database connection string. Storing the connection string in the Web.config file makes sense, since any modifications to the database configurations can be maintained at a single location. As otherwise we'll have to keep it either as a class level variable in the entire associated source files or keep it in another class as a public static variable.



```
<?xml version="1.0"?>
<!-- For more information on how to configure your ASP.NET application, please visit
      http://go.microsoft.com/fwlink/?linkid=169433
-->
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" >
      <assemblies>
        <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
      </assemblies>
    </compilation>
    <httpRuntime requestValidationMode="2.0" />
  </system.web>
  <appSettings>
    <add key="ConnectionString" value="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver"/>
    <!--<add key="ClientId" value="WKCbVew01"/>
    <add key="ClientSecret" value="BEBcSw005k08BCZTMncjkwD34xUXJqGxHBlHSYq8gZal147"/>
    <add key="RedirectUrl" value="http://ec2-50-16-101-227.compute-1.amazonaws.com/Search.aspx"/>-->
    <add key="ClientId" value="QRie5QVOEn"/>
    <add key="ClientSecret" value="jBDRFY225mDMky1Ywj1VgHvUPsiDWOTcxwqvyc4zCACaON2"/>
    <add key="RedirectUrl" value="http://localhost:3290/Search.aspx"/>
    <add key="SlcApiUrl" value="https://api.sandbox.slcedu.org/api/rest/v1"/>
    <add key="OauthUrl" value="https://api.sandbox.slcedu.org/api/oauth"/>
    <add key="UserSession" value="https://api.sandbox.slcedu.org/api/rest/system/session/check"/>
  </appSettings>
  <connectionStrings>
    <add name="SLC_SDACEntities" connectionString="metadata=res://*/SDACEntityModel.csdl|res://*/SDACEntityModel.ssdl|res://*/SDACEntityModel.msdl;provider=System.Data.SqlClient;provider connection string="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver;MultipleActiveResultSets=True"; providerName="System.Data.EntityClient" />
    <add name="SDACConnectionString" connectionString="Data Source=DELL-PC;Initial Catalog=SDAC;Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

Connection String



```
<?xml version="1.0"?>
<!-- For more information on how to configure your ASP.NET application, please visit
      http://go.microsoft.com/fwlink/?linkid=169433
-->
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" >
      <assemblies>
        <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
      </assemblies>
    </compilation>
    <httpRuntime requestValidationMode="2.0" />
  </system.web>
  <appSettings>
    <add key="ConnectionString" value="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver"/>
    <!--<add key="ClientId" value="WKCbVew01"/>
    <add key="ClientSecret" value="BEBcSw005k08BCZTMncjkwD34xUXJqGxHBlHSYq8gZal147"/>
    <add key="RedirectUrl" value="http://ec2-50-16-101-227.compute-1.amazonaws.com/Search.aspx"/>-->
    <add key="ClientId" value="QRie5QVOEn"/>
    <add key="ClientSecret" value="jBDRFY225mDMky1Ywj1VgHvUPsiDWOTcxwqvyc4zCACaON2"/>
    <add key="RedirectUrl" value="http://localhost:3290/Search.aspx"/>
    <add key="SlcApiUrl" value="https://api.sandbox.slcedu.org/api/rest/v1"/>
    <add key="OauthUrl" value="https://api.sandbox.slcedu.org/api/oauth"/>
    <add key="UserSession" value="https://api.sandbox.slcedu.org/api/rest/system/session/check"/>
  </appSettings>
  <connectionStrings>
    <add name="SLC_SDACEntities" connectionString="metadata=res://*/SDACEntityModel.csdl|res://*/SDACEntityModel.ssdl|res://*/SDACEntityModel.msdl;provider=System.Data.SqlClient;provider connection string="Data Source=DELL-PC;Initial Catalog=SDAC;User ID=sa;Password=sqlserver;MultipleActiveResultSets=True"; providerName="System.Data.EntityClient" />
    <add name="SDACConnectionString" connectionString="Data Source=DELL-PC;Initial Catalog=SDAC;Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

Changes in Connection String

- a. **Data Source**
Set Data Source = Server name i.e. Here it is DELL-PC
- b. **Initial Catalog**
Set Initial Catalog = Your database name
- c. **Password**
Set Password = Your SQL Server password.

