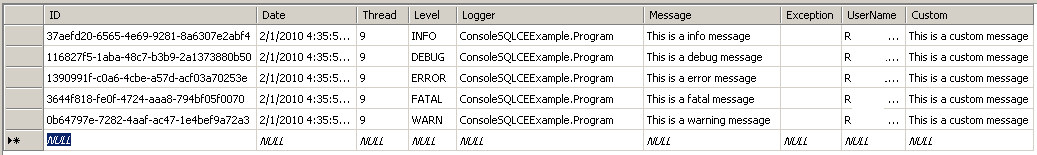
[**Log4net with SQL CE (Compact Edition)**](http://www.npcompetesolutions.com/Blogs/?p=3)

February 1, 2010, 5:35 am

The following guide shows how to use Sql Server CE (Compact Edition) to log your application events with Log4net.  
Your SQL Server CE log table will look like this:



[Click here](http://www.npcompetesolutions.com/Blogs/wp-content/uploads/Log4NetSQLCE/ConsoleSQLCEExample.zip) to download a complete version of this application.

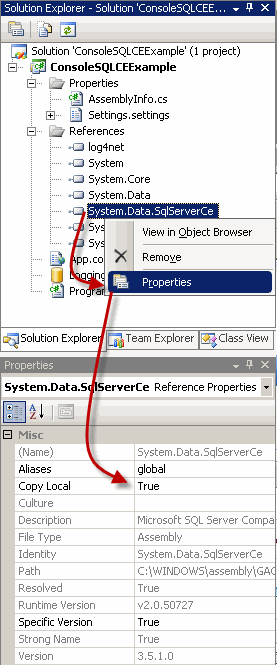
*Note: This tutorial references* [***log4net***](http://logging.apache.org/log4net/index.html) *version 1.2.10 and* [***SQL Server CE***](http://www.microsoft.com/Sqlserver/2008/en/us/compact.aspx) *version 3.5.*

1. Create a new Console application solution.

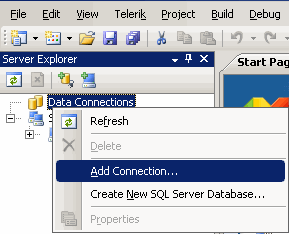
2. Add to the project a reference to the [...]\bin\net\2.0\release\log4net.dll assembly in the log4net distribution.

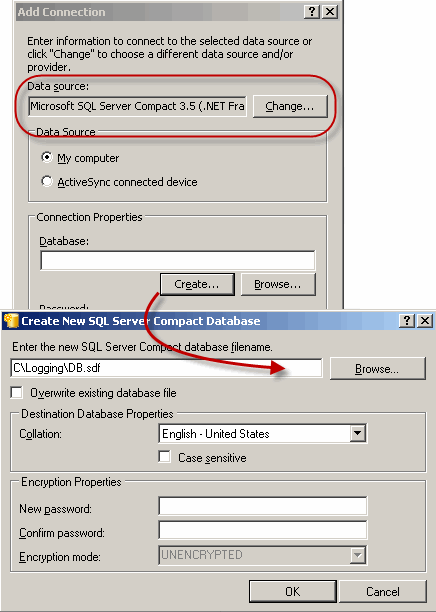
3. Add to the project a reference to the SQL Server CE assembly: [...]\Program Files\Microsoft SQL Server Compact Edition\v3.5\Desktop\System.Data.SqlServerCe.Dll.

IMPORTANT! Make sure you set “Copy Local” to true for this dll.



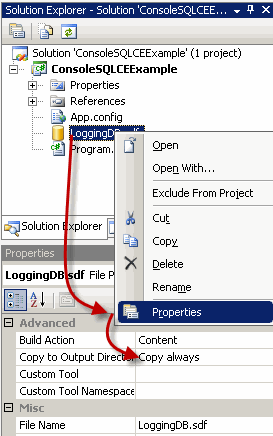
4. Create logging database. Server Explorer – > Right click on Data Connections -> Add Connection -> Data Source should be “Microsoft SQL Server Compact 3.5 (.NET Framework Data Provider for Microsoft SQL Server Compact 3.5)” -> Click “Create” button -> Enter the new SQL Server Conpact database filename “LoggingDB.sdf” -> Click Ok -> Select “Yes” to continue with the blank password. – > Click Ok to close the “Add Connection” dialog.



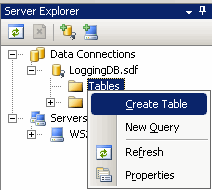


5. Attach the LoggingDB.sdf file to the project (right click on the project name -> Add -> Existing Item -> Browse to LoggingDB.sdf file)

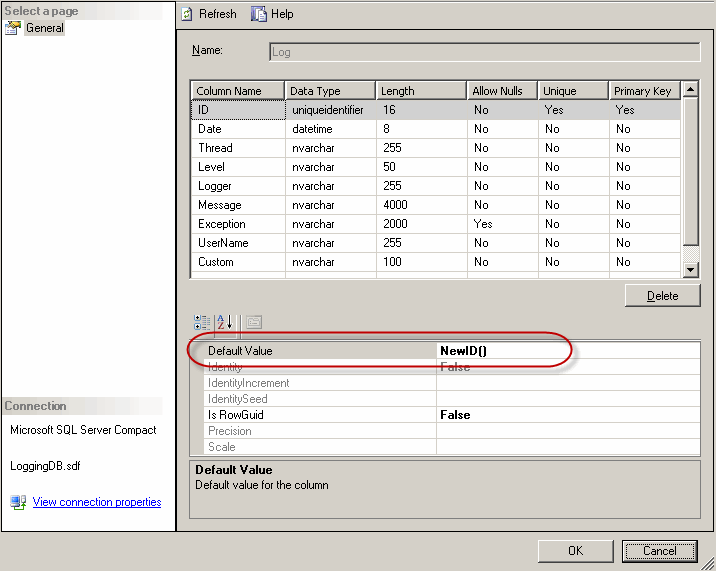
6. Set “Copy to Output Directory” propertyes to “Copy always” (right Click on LoggingDB.asf -> Properties -> Select “Copy Always” from the option list)



7. Create a Log table in the data base )Right click on Tables -> Create Table)



8. The table schema: (make sure you have NewID() as default value for ID field)



9. Add the following application configuration file (App.Config) to the console project:

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<configSections>

<section name="log4net"

type="log4net.Config.Log4NetConfigurationSectionHandler,Log4net"/>

</configSections>

<log4net>

<root>

<level value="DEBUG" />

<appender-ref ref="SQLCEAppender" />

</root>

<appender name="SQLCEAppender" type="log4net.Appender.AdoNetAppender">

<connectionType

value="System.Data.SqlServerCe.SqlCeConnection, System.Data.SqlServerCe" />

<connectionString value="Data Source='LoggingDB.sdf';" />

<commandText

value="INSERT INTO Log

([Date],[Thread],[Level],[Logger],[Message],

[Exception], [UserName], [Custom])

VALUES

(@log\_date, @thread, @log\_level, @logger, @message,

@exception, @username, @custom)" />

<parameter>

<parameterName value="@log\_date" />

<dbType value="DateTime" />

<layout type="log4net.Layout.RawUtcTimeStampLayout" />

</parameter>

<parameter>

<parameterName value="@thread" />

<dbType value="String" />

<size value="255" />

<layout type="log4net.Layout.PatternLayout">

<conversionPattern value="%thread" />

</layout>

</parameter>

<parameter>

<parameterName value="@log\_level" />

<dbType value="String" />

<size value="50" />

<layout type="log4net.Layout.PatternLayout">

<conversionPattern value="%level" />

</layout>

</parameter>

<parameter>

<parameterName value="@logger" />

<dbType value="String" />

<size value="255" />

<layout type="log4net.Layout.PatternLayout">

<conversionPattern value="%logger" />

</layout>

</parameter>

<parameter>

<parameterName value="@message" />

<dbType value="String" />

<size value="4000" />

<layout type="log4net.Layout.PatternLayout">

<conversionPattern value="%message" />

</layout>

</parameter>

<parameter>

<parameterName value="@exception" />

<dbType value="String" />

<size value="2000" />

<layout type="log4net.Layout.ExceptionLayout" />

</parameter>

<parameter>

<parameterName value="@username" />

<dbType value="String" />

<size value="255" />

<layout type="log4net.Layout.PatternLayout">

<conversionPattern value="%username" />

</layout>

</parameter>

<parameter>

<parameterName value="@custom" />

<dbType value="String" />

<size value="255" />

<layout type="log4net.Layout.PatternLayout">

<conversionPattern value="%property{Custom}" />

</layout>

</parameter>

</appender>

</log4net>

</configuration>

10. Modify your Program.cs file:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using log4net;

using log4net.Config;

namespace ConsoleSQLCEExample

{

class Program

{

private static log4net.ILog Log =

log4net.LogManager.GetLogger(typeof(Program));

static void Main(string[] args)

{

log4net.Config.XmlConfigurator.Configure();

log4net.ThreadContext.Properties["Custom"] =

"This is a custom message";

Log.Info("This is a info message");

Log.Debug("This is a debug message");

Log.Error("This is a error message");

Log.Fatal("This is a fatal message");

Log.Warn("This is a warning message");

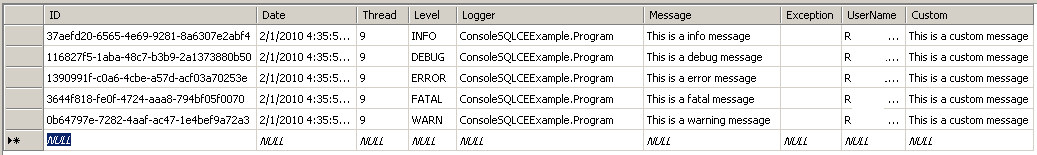
}

}

}

11. Run the application.  
12. Connect to LoggingDB.sdf in you \bin\Debug (or Releaase) folder (You can use Server Explorer -> Right click on Data Connetions -> Add Connection -> Click on “Browse” button and navigate to you \bin\debug\LoggingDB.sdf file)

13. Each log message produced by the utility routine will be saved to the Log table:



**Coming Up**

Next time we will add funtionality that sends the logg information over internet to a central server using Microsoft Sync Framework.  
This way we will have multiple desktop application logs on a central place.

Thank you,  
[**npCompeteSolutions.com**](http://npcompetesolutions.com/)