## 【附加数据库】

#region 附加数据库

try

{

//数据库连接对象

using (SqlConnection Conn = new SqlConnection("Data Source=.;Persist Security Info=False;User ID=sa; pwd =sa"))

{

using (SqlCommand Comm = new SqlCommand())//命令

{

Comm.CommandText = "sp\_attach\_db";

Comm.CommandType = CommandType.StoredProcedure; //引入存储过程

Comm.Parameters.Add("@dbname", "STBInfo"); //这个是数据库名字，也就是你写什么名字附加到数据库就是什么名字。

Comm.Parameters.Add("@filename1", path + @"\App\_Data\STBInfo.mdf"); //地址

Comm.Parameters.Add("@filename2", path + @"\App\_Data\STBInfo\_log.ldf");

Comm.Connection = Conn; //初始化连接对象

Conn.Open(); //打开连接对象

Comm.ExecuteNonQuery();

Conn.Close(); //关闭连接对象

}

}

}

catch (Exception ex)

{

MessageBox.Show("附加失败:" + ex.Message);

}

#endregion

## 【分离数据库】

#region 分离数据库

private void btn\_detach\_Click(object sender, EventArgs e)

{

using (SqlConnection Conn = new SqlConnection(ConfigurationManager.ConnectionStrings["connStringName"].ConnectionString))

{

using (SqlCommand Comm = new SqlCommand())//命令

{

try

{

Conn.Open();

Comm.Connection = Conn;

Comm.CommandText = @"USE master;

ALTER DATABASE STBInfo

SET SINGLE\_USER

with ROLLBACK IMMEDIATE";

//上面的目的是强制断开用户

Comm.ExecuteNonQuery();

Comm.CommandText = @"sp\_detach\_db";

Comm.Parameters.Add(new SqlParameter(@"dbname", SqlDbType.NVarChar));

Comm.Parameters[@"dbname"].Value = dataBaseName;

Comm.CommandType = CommandType.StoredProcedure;

Comm.ExecuteNonQuery();

HDIC\_Message.ShowInfoDialog(this, "分离数据库成功");

}

catch (System.Exception ex)

{

MessageBox.Show("分离失败:" + ex.Message);

}

finally

{

Conn.Close();

}

}

}

}

#endregion

## 【备份数据库】

#region 备份数据库

private void btn\_BackUpDB\_Click(object sender, EventArgs e)

{

FolderBrowserDialog fbDialog = new FolderBrowserDialog();

if (fbDialog.ShowDialog() == DialogResult.OK)

{

string backupstr = "backup database " + dataBaseName + " to disk='" + fbDialog.SelectedPath + "\\"+dataBaseName+".bak';";

using (SqlConnection Conn = new SqlConnection(ConfigurationManager.ConnectionStrings["connStringName"].ConnectionString))

{

using (SqlCommand Comm = new SqlCommand(backupstr, Conn))//命令

{

try

{

Conn.Open();

Comm.ExecuteNonQuery();

HDIC\_Message.ShowInfoDialog(this, "数据库备份成功");

}

catch (Exception ex)

{

HDIC\_Message.ShowInfoDialog(this, "数据库备份失败\n" + ex.ToString());

}

finally

{

Conn.Close();

}

}

}

}

}

#endregion

## 【还原数据库】

#region 还原数据库

private void btn\_Restore\_Click(object sender, EventArgs e)

{

OpenFileDialog fbDialog = new OpenFileDialog();

if (fbDialog.ShowDialog() == DialogResult.OK)

{

using (SqlConnection Conn = new SqlConnection())

{

using (SqlCommand Comm = new SqlCommand())//命令

{

try

{

Conn.ConnectionString = "Data Source=.;Initial Catalog="+dataBaseName+";Persist Security Info=False;User ID=sa; pwd =sa";

Comm.Connection = Conn;

Conn.Open();

#region 第一种数据库还原方法

//先杀所用访问该数据库进程

string strSQL = "use master ;select spid from master..sysprocesses where dbid=db\_id( '" + dataBaseName + "') ";//获取所有用户进程

SqlDataAdapter Da = new SqlDataAdapter(strSQL, Conn);

DataTable spidTable = new DataTable();

Da.Fill(spidTable);

Comm.CommandType = CommandType.Text;

for (int iRow = 0; iRow <= spidTable.Rows.Count - 1; iRow++)

{

Comm.CommandText = "kill " + spidTable.Rows[iRow][0].ToString(); //强行关闭用户进程

Comm.ExecuteNonQuery();

}

//还原指定的数据库文件

Comm.CommandText = "restore database " + dataBaseName + " from disk='" + fbDialog.FileName + "'";

Comm.ExecuteNonQuery();

#endregion

#region 第二种数据库还原方法

////先强制关闭用户进程，然后还原指定的数据库文件

string sql = string.Format("use master ;declare @s varchar(8000);select @s=isnull(@s,'')+' kill '+rtrim(spID) from master..sysprocesses where dbid=db\_id('{0}');select @s;exec(@s) ;RESTORE DATABASE {1} FROM DISK = N'{2}' with replace", dataBaseName, dataBaseName, fbDialog.FileName);

Comm.CommandText = sql;

Comm.CommandType = CommandType.Text;

Comm.ExecuteNonQuery();

#endregion

HDIC\_Message.ShowInfoDialog(this, "恢复成功");

}

catch (Exception ex)

{

HDIC\_Message.ShowInfoDialog(this, "恢复失败\n" + ex.ToString());

}

finally

{

Conn.Close();

}

}

}

}

}

#endregion