using System.Data;

using System.Data.SqlClient;

using System.Collections.Generic;

using DataAccessLayer;

namespace BusinessLayer

{

public class Customer : ObjectBase

{

public static string OBJECT\_NAME = "Customer";

public string Name = "";

public string Address = "";

public string Mobile = "";

public string Phone = "";

public string Fax = "";

public string Email = "";

public string Homepage = "";

public string Username = "";

public string Password = "";

public string Country = "";

public int CountryID = NullHelper.Integer;

public int Active = 1;

public Customer()

{ SetObjectName(OBJECT\_NAME); }

public Customer(int objectID) : this()

{ ID = objectID; }

protected override bool Reset()

{

Name = "";

Address = "";

Mobile = "";

Phone = "";

Fax = "";

Email = "";

Homepage = "";

Username = "";

Password = "";

Country = "";

CountryID = NullHelper.Integer;

Active = 1;

return base.Reset();

}

protected override bool Read(DataRow dataRow)

{

Name = Read<string>(dataRow, "Name", NullHelper.String);

Address = Read<string>(dataRow, "Address", NullHelper.String);

Mobile = Read<string>(dataRow, "Mobile", NullHelper.String);

Phone = Read<string>(dataRow, "Phone", NullHelper.String);

Fax = Read<string>(dataRow, "Fax", NullHelper.String);

Email = Read<string>(dataRow, "Email", NullHelper.String);

Homepage = Read<string>(dataRow, "Homepage", NullHelper.String);

Country = Read<string>(dataRow, "Country", NullHelper.String);

CountryID = Read<int>(dataRow, "CountryID", NullHelper.Integer);

Active = Read<int>(dataRow, "Active", 1);

return base.Read(dataRow);

}

protected override bool Populate()

{

bool Ret = false;

DataRow row = Populate("");

if (null != row && row.ItemArray.Length > 0)

Ret = Read(row);

return Ret;

}

public override int Save()

{

int Ret = 0;

int paramSize = 13;

EnumProcType procType = EnumProcType.Insert;

ParameterDirection paramDir = ParameterDirection.Output;

AdaptForSave(ref procType, ref paramDir, ref paramSize);

SqlParameter[] procParams = new SqlParameter[paramSize]; int i = 0;

procParams[i] = DataAccess.CreateParameter("@ID", SqlDbType.Int, ID, paramDir);

if (EnumProcType.Insert == procType)

procParams[++i] = DataAccess.CreateParameter("@CreateDate", SqlDbType.DateTime, CreateDate, ParameterDirection.Output);

procParams[++i] = DataAccess.CreateParameter("@ModifyDate", SqlDbType.DateTime, ModifyDate, ParameterDirection.Output);

procParams[++i] = DataAccess.CreateParameter("@Name", SqlDbType.NVarChar, Name);

procParams[++i] = DataAccess.CreateParameter("@Address", SqlDbType.NVarChar, Address);

procParams[++i] = DataAccess.CreateParameter("@Mobile", SqlDbType.NVarChar, Mobile);

procParams[++i] = DataAccess.CreateParameter("@Phone", SqlDbType.NVarChar, Phone);

procParams[++i] = DataAccess.CreateParameter("@Fax", SqlDbType.NVarChar, Fax);

procParams[++i] = DataAccess.CreateParameter("@Email", SqlDbType.NVarChar, Email);

procParams[++i] = DataAccess.CreateParameter("@Homepage", SqlDbType.NVarChar, Homepage);

procParams[++i] = DataAccess.CreateParameter("@Username", SqlDbType.NVarChar, Username);

procParams[++i] = DataAccess.CreateParameter("@Password", SqlDbType.NVarChar, Password);

procParams[++i] = DataAccess.CreateParameter("@CountryID", SqlDbType.Int, CountryID);

procParams[++i] = DataAccess.CreateParameter("@Active", SqlDbType.Int, Active);

Ret = Save(procType, ref procParams);

return Ret;

}

public override void Dispose()

{ base.Dispose(); }

}

}