public static DataTable GetEmployeesFromFieldGlass(DataTable attuids)

{

DataTable dt = new DataTable();

SqlConnection connection = null;

SqlCommand command = null;

SqlDataAdapter adapter = null;

try

{

string connStr = ConfigurationManager.ConnectionStrings["ATTUID"].ConnectionString;

connection = new SqlConnection(connStr);

command = new SqlCommand("GetEmployees", connection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@ATTUIDs", SqlDbType.Structured).Value = attuids;

adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

}

catch (Exception ex)

{

log.Error(ex);

}

finally

{

if (adapter != null) adapter.Dispose();

if (command != null) command.Dispose();

if (connection != null)

{

if (connection.State != ConnectionState.Closed)

connection.Close();

connection.Dispose();

}

}

return dt;

}

public static DataTable GetRecords(string filename)

{

using (SqlConnection sqlcon = new

SqlConnection(ConfigurationManager.ConnectionStrings["SqlConn"].ConnectionString))

{

using (SqlCommand cmd = new SqlCommand("GetDaily", sqlcon))

{

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@FileName", filename);

using (SqlDataAdapter da = new SqlDataAdapter(cmd))

{

DataTable dt = new DataTable();

da.Fill(dt);

return dt;

}

}

}

}

public static void UpdateDB(string strATTUID, string strEMPID)

{

string connStr = ConfigurationManager.ConnectionStrings["ATTUID"].ConnectionString;

SqlCommand command = null;

SqlConnection connection = new SqlConnection(connStr);

try

{

command = new SqlCommand(@"ATT\_SetModifiedDates", connection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add("@EMPLID", SqlDbType.VarChar).Value = strEMPID;

command.Parameters.Add("@ModifiedOn", SqlDbType.DateTime).Value = DateTime.Now;

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

catch (Exception ex)

{

log.Error(ex);

}

finally

{

if (command != null) command.Dispose();

if (connection != null)

{

if (connection.State != ConnectionState.Closed)

connection.Close();

connection.Dispose();

}

}

}

ALTER PROCEDURE [dbo].[ATT\_SetModifiedDates]

@EMPLID [nvarchar](255),

@NotificationSentOn [datetime] = NULL

AS

BEGIN

SET NOCOUNT ON;

update [tblDailyReport2]

set

NotificationSentOn = case when @NotificationSentOn is null then NotificationSentOn

else @NotificationSentOn end,

AddedOn = case when @AddedOn is null then AddedOn

else @AddedOn end,

where EMPLID = @EMPLID

and ATTUID = @ATTUID

END