UPTDATE datresult with WorkStatus from other table

GENERATE A TABLW WITH ATTUID , WorkStatus

Create a list, with all ATTUID

var attuids = from i in dataresult

select new

{

i.ATTUID

};

RETRIEVE A TABLE WITH ALL THE ATTUIDS.

DataTable employees = GetEmployeesFromFieldGlass(ConvertEnumToDataTable(attuids));

public static DataTable GetEmployeesFromFieldGlass(DataTable attuids)

{

DataTable dt = new DataTable();

SqlConnection connection = null;

SqlCommand command = null;

SqlDataAdapter adapter = null;

try

{

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

connection = new SqlConnection(connectionstrings);

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

command.CommandType = CommandType.StoredProcedure;

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

adapter = new SqlDataAdapter(command);

adapter.Fill(dt);

}

catch (Exception 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;

}

//

ALTER PROCEDURE [dbo].[GetEmployeesFromFieldGlass]

@ATTUIDs StringIdTableType readonly

AS

BEGIN

SET NOCOUNT ON;

select

lower(ATTUID) ATTUID,

WorkOrder\_ID\_Status,

from vFieldGlassReport t1

where ATTUID in (select distinct ID from @ATTUIDs)

and END\_DATE = (SELECT MAX(END\_DATE) from vFieldGlassReport

where ATTUID = t1.ATTUID )

order by ATTUID

END

USING JOIN

var data2 = from d in dataresult.AsEnumerable()

join r in employees.AsEnumerable()

on d.ATTUID.ToLower() equals r.Field<string>("ATTUID").ToLower()

into rt

from join1 in rt.DefaultIfEmpty()

select new

{

EmployeeId = d.EmployeeID,

WorkStatus = (join1 == null ? string.Empty : join1.Field<string>("WorkStatus")),

Comments = d.Comments,

Location = d.Location

};

DataGridView dg = new DataGridView();

DataTable dt = ConvertEnumToDataTable(dataresult2);

DataGrid.DataSource = dt;

OTRA FORMA

public class ATTUIDS

{

public string ATTUID { get; set; }

}

List<ATTUIDS> attuis = new List<ATTUIDS>();

attuis = itemList.Select(x =>

{

string \_ATTUID = x.FieldValues[fieldmap["ATTUID"].FirstOrDefault()] == null ? "" :

x.FieldValues["ATTUID"].ToString();

return new ATTUIDS

{

ATTUID = \_ATTUID

};

}).ToList();

DataTable employees2 = GetEmployeesFromFieldGlass(ConvertEnumToDataTable(attuis));

List<DataResultItem> dataList = ( from d in itemList.AsEnumerable()

join r in employees2.AsEnumerable()

on d.FieldValues["ATTUID"] equals r.Field<string>("ATTUID").ToLower()

into rt

from join1 in rt.DefaultIfEmpty()

select new DataResultItem

{

ATTUID = d.FieldValues["ATTUID"] == null ? "" : d.FieldValues["ATTUID"].ToString(),

Comments = d.FieldValues["Comments"] == null ? "" : d.FieldValues["Comments"].ToString(),

EmployeeID = d.FieldValues["Employee ID"].ToString(),

Requestor = ((FieldUserValue)d.FieldValues[Requestor"].FirstOrDefault()]).LookupValue,

OrderStatus = (join1 == null ? string.Empty : join1.Field<string>("WorkOrder\_ID\_Status"))

}).ToList();

DataGrid.DataSource = dataList;

// PARA QUE PUEDA DARLE SORT

DataGrid.DataSource = ConvertEnumToDataTable(dataList);

OTRA FORMA

EN VEZ DE JOIN, HACER UN LOOKUP

public class DataResultItem

{

public string EmployeeID { get; set; }

public string ATTUID { get; set; }

public string EmployeeName { get; set; }

public string Comments { get; set; }

public string Requestor { get; set; }

}

public class ATTUIDS

{

public string ATTUID { get; set; }

}

// GET ALL ATTUID’s

List<ATTUIDS> attuis = new List<ATTUIDS>();

attuis = itemList.Select(x =>

{

string \_ATTUID = x.FieldValues["ATTUID"] == null ? "" : x.FieldValues["ATTUID"].ToString();

return new ATTUIDS

{

ATTUID = \_ATTUID

};

}).ToList();

// CREATE TABLE TO LOOKUP

DataTable employees = GetEmployeesFromFieldGlass(ConvertEnumToDataTable(attuis));

List<DataResultItem> dataResult = new List<DataResultItem>();

dataResult = itemList.Select(x =>

{

string \_ATTUID = d.FieldValues["ATTUID"] == null ? "" : d.FieldValues["ATTUID"].ToString(),

string \_Comments = d.FieldValues["Comments"] == null ? "" : d.FieldValues["Comments"].ToString(),

string \_Requestor = ((FieldUserValue)d.FieldValues[Requestor"].FirstOrDefault()]).LookupValue

// LOOKUP TO AVOID JOIN

String \_OrderStatus = "";

if (\_ATTUID != "")

{

string where1 = "ATTUID = '" + \_ATTUID + "'";

DataRow[] result = employees.Select(where1);

\_OrderStatus = result.Length > 0 ? result[0]["WorkOrder\_ID\_Status"].ToString() : "";

}

return new DataResultItem

{

EmployeeID = x.FieldValues["Employee ID"].ToString(),

WorkOrder = \_workOrder,

WorkOrderStatus = \_OrderStatus,

ATTUID = \_ATTUID,

Comments = \_Comments,

Requestor = \_requestor

};

}).ToList();

// DataGridView sorts DataTables, but not Lists<T>

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DataGrid.DataSource = ConvertEnumToDataTable(dataResult);

ConvertEnumToDataTable

using System.Reflection;

private DataTable ConvertEnumToDataTable(IEnumerable ien)  
{  
 DataTable dt = new DataTable();  
 foreach (object obj in ien)  
 {  
 Type t = obj.GetType();  
 PropertyInfo[] pis = t.GetProperties();

if (dt.Columns.Count == 0)  
 {  
 foreach (PropertyInfo pi in pis)  
 {  
 dt.Columns.Add(pi.Name, pi.PropertyType);  
 }  
 }  
 DataRow dr = dt.NewRow();

foreach (PropertyInfo pi in pis)  
 {  
 object value = pi.GetValue(obj, null);  
 dr[pi.Name] = value;  
 }  
 dt.Rows.Add(dr);  
 }  
 return dt;  
}