public class ShortenEndDate

{

public string EmployeeID { get; set; }

public string EmpFirstName { get; set; }

public string EmpLastName { get; set; }

public string ATTUID { get; set; }

public string EffectiveDateOfAttrition { get; set; }

public string AttritionReason { get; set; }

public string SMSNumber { get; set; }

public string AutomationRequestor { get; set; }

public string RequestorEmail { get; set; }

public string Company { get; set; }

public string Status { get; set; }

public string Comments { get; set; }

public string WorkOrder\_ID { get; set; }

public string Employee\_Name { get; set; }

public string ElapsedTime { get; set; }

//http://stackoverflow.com/questions/709560/linq-in-line-property-update-during-join

public DeactivationItem UpdateWithFieldGlass(DataRow r)

{

WorkOrder\_ID = (r == null ? string.Empty : r.Field<string>("WorkOrder\_ID"));

Employee\_Name = (r == null ? string.Empty : r.Field<string>("Employee\_Name"));

return this;

}

}

static List<DeactivationItem> GetItems()

{

int requested = 0;

List<DeactivationItem> deactivationItems = new List<DeactivationItem>();

ClientContext context = null;

try

{

context = new ClientContext(SiteUrl);

List list = context.Web.Lists.GetByTitle(ListTitle);

//get column names

FieldCollection fields;

List<Field> fieldList;

Field field;

fields = list.Fields;

context.Load(fields, collection => collection.Include(f => f.Title, f => f.InternalName));

context.ExecuteQuery();

fieldList = fields.ToList();

foreach (string column in listColumns)

{

field = fieldList.FirstOrDefault(f => f.Title == column);

if (field != null)

columns.Add(field.Title, field.InternalName); //resolve Field Internal Name by Title

}

//get items

CamlQuery query = new CamlQuery();

string viewXml = "<View><Query><Where><And>COMPANYCONDITION<Or><Or><Eq><FieldRef Name='Status' /><Value Type='Choice'>Requested</Value></Eq><Eq><FieldRef Name='Status' /><Value Type='Choice'>Processing</Value></Eq></Or><And><Eq><FieldRef Name='Status' /><Value Type='Choice'>Error</Value></Eq><Contains><FieldRef Name='Comments' /><Value Type='Text'>Exception occurred in FieldGlass. Will retry in next run.</Value></Contains></And></Or></And></Where><OrderBy><FieldRef Name='Author' Ascending='True' /></OrderBy></Query></View>";

if (string.IsNullOrEmpty(Company))

viewXml = viewXml.Replace("COMPANYCONDITION", "<IsNull><FieldRef Name='Company' /></IsNull>");

else

viewXml = viewXml.Replace("COMPANYCONDITION", "<Eq><FieldRef Name='Company' /><Value Type='Text'>" + Company + "</Value></Eq>");

query.ViewXml = viewXml;

ListItemCollection items = list.GetItems(query);

context.Load(items);

context.ExecuteQuery();

requested = items.Count;

int count = 0;

foreach (ListItem item in items)

{

DeactivationItem deactivationitem = new DeactivationItem();

//validate each item

if (item[columns["ATTUID"]] == null)

{

item[columns["Status"]] = "Error";

item[columns["Comments"]] = "ATTUID is required";

}

else

{

deactivationitem.EmployeeID = item[columns["Employee ID"]].ToString().Trim();

deactivationitem.EmpFirstName = item[columns["Emp First Name"]].ToString().Trim();

deactivationitem.EmpLastName = item[columns["Emp Last Name"]].ToString().Trim();

deactivationitem.ATTUID = item[columns["ATTUID"]].ToString().ToLower().Trim();

deactivationitem.EffectiveDateOfAttrition = ((DateTime)item[columns["Effective Date of Attrition"]]).ToString("MM/dd/yyyy");

deactivationitem.AttritionReason = item[columns["Attrition Reason"]].ToString().Trim();

deactivationitem.SMSNumber = item[columns["SMS Number"]].ToString().Trim();

var requestor = context.Web.EnsureUser(((FieldUserValue)item[columns["Automation Requestor"]]).LookupValue);

context.Load(requestor);

context.ExecuteQuery();

deactivationitem.AutomationRequestor = requestor.Title;

deactivationitem.RequestorEmail = requestor.Email;

deactivationitem.Status = "Processing";

deactivationitem.Comments = string.Empty;

deactivationItems.Add(deactivationitem);

//columns to update in sharepoint

item[columns["Employee ID"]] = deactivationitem.EmployeeID;

item[columns["Emp First Name"]] = deactivationitem.EmpFirstName;

item[columns["Emp Last Name"]] = deactivationitem.EmpLastName;

item[columns["ATTUID"]] = deactivationitem.ATTUID;

item[columns["Attrition Reason"]] = deactivationitem.AttritionReason;

item[columns["SMS Number"]] = deactivationitem.SMSNumber;

item[columns["Status"]] = deactivationitem.Status;

item[columns["Comments"]] = deactivationitem.Comments;

}

item.Update();

count++;

if (count % BatchSize == 0 || count == requested)

context.ExecuteQuery();

}

if (deactivationItems.Count > 0)

{

//get WorkOrderId, Employee Name from Database

var attuids = from i in deactivationItems

select new

{

i.ATTUID

};

DataTable employees = DataAccess.GetEmployeesFromFieldGlass(attuids.CopyToDataTable());

var match = from d in deactivationItems

join e in employees.AsEnumerable()

on d.ATTUID equals e.Field<string>("ATTUID") into de

from r in de.DefaultIfEmpty()

select d.UpdateWithFieldGlass(r);

var array = match.ToArray(); //needed only for the code above to take effect, don't remove!

}

}

catch (Exception ex)

{

log.Error(ex.ToString());

}

finally

{

if (context != null)

context.Dispose();

}

return deactivationItems;

}

= = = = = = =

where

< CONDITION >

And

  (

      status = Requested

      Or status = Processing

      Or ( status = Error and comment = "aaaa" )

   )

order by AUthor asc

  string viewXml = "<View><Query><Where>

<And>COMPANYCONDITION

<Or>

<Or>

<Eq><FieldRef Name='Status' /><Value Type='Choice'>Requested</Value></Eq>

<Eq><FieldRef Name='Status' /><Value Type='Choice'>Processing</Value></Eq>

</Or>

<And><Eq><FieldRef Name='Status' /><Value Type='Choice'>Error</Value></Eq>

<Contains><FieldRef Name='Comments' /><Value Type='Text'>Exception occurred in FieldGlass. Will retry in next run.</Value></Contains>

</And>

</Or>

</And>

</Where>

<OrderBy><FieldRef Name='Author' Ascending='True' /></OrderBy></Query></View>";