* Receives a DataTable
* Takes the distro
* Set the subject and body
* Creates an excel file
* But first deletes any previous file with the same name
* Send an email with the path where the excel file is to be attached

void Acknowledge(DataTable table)

{

string sender = ConfigurationManager.AppSettings["TASK11"].ToString();

FILENAME = "C:\\temp\\NewCases.xlsx";

Exposure.Program.ExportData(table, FILENAME);

string[] File = new string[1];

File[0] = FILENAME;

Exposure.SendEmails send = new Exposure.SendEmails();

string subject = "Confirmed Cases for Medical Assessment";

send.GeneralEmail(subject, "Acknowledge", File, sender);

}

static public void ExportData(DataTable table, string path)

{

//string script = "window.open('" + pageurl + "','\_blank');";

//ScriptManager.RegisterStartupScript(this, GetType(), pageurl, script, true);

log.Info(path);

using (var workbook = SpreadsheetDocument.Create(path, DocumentFormat.OpenXml.SpreadsheetDocumentType.Workbook))

{

var workbookPart = workbook.AddWorkbookPart();

workbook.WorkbookPart.Workbook = new DocumentFormat.OpenXml.Spreadsheet.Workbook();

workbook.WorkbookPart.Workbook.Sheets = new DocumentFormat.OpenXml.Spreadsheet.Sheets();

var sheetPart = workbook.WorkbookPart.AddNewPart<WorksheetPart>();

var sheetData = new DocumentFormat.OpenXml.Spreadsheet.SheetData();

sheetPart.Worksheet = new DocumentFormat.OpenXml.Spreadsheet.Worksheet(sheetData);

Sheets sheets = workbook.WorkbookPart.Workbook.GetFirstChild<DocumentFormat.OpenXml.Spreadsheet.Sheets>();

string relationshipId = workbook.WorkbookPart.GetIdOfPart(sheetPart);

Sheet sheet = new Sheet() { Id = relationshipId, SheetId = 1, Name = "Sheet 1" };

sheets.Append(sheet);

Row headerRow = new DocumentFormat.OpenXml.Spreadsheet.Row();

List<String> columns = new List<string>();

foreach (DataColumn column in table.Columns)

{

columns.Add(column.ColumnName);

Cell cell = new Cell();

cell.DataType = CellValues.String;

cell.CellValue = new CellValue(column.ColumnName);

headerRow.AppendChild(cell);

}

log.Info("Header created");

sheetData.AppendChild(headerRow);

foreach (DataRow dsrow in table.Rows)

{

Row newRow = new Row();

int conteo = 0;

foreach (String col in columns)

{

Cell cell = new Cell();

cell.DataType = CellValues.String;

try

{

cell.CellValue = new CellValue(dsrow[col].ToString());

}

catch(Exception )

{

log.Info("Error " + conteo.ToString());

cell.CellValue = new CellValue("");

}

newRow.AppendChild(cell);

conteo++;

}

sheetData.AppendChild(newRow);

}

}

}

public void GeneralEmail(string subject, string body, string[] File, string sender)

{

SmtpClient client = null;

MailMessage message = null;

try

{

client = new SmtpClient(ConfigurationManager.AppSettings["SmtpServer"].ToString());

client.UseDefaultCredentials = false;

log.Info("server");

message = new MailMessage();

message.From = new MailAddress(ConfigurationManager.AppSettings["EmailFrom"].ToString());

message.Subject = subject;

log.Info(subject + " " + message.From);

string body2 = "";

if (sender == "") sender = ConfigurationManager.AppSettings["EmailTo"].ToString();

#if DEBUG

body2 = "Sent in PROD to " + sender + "</br></br>";

sender = ConfigurationManager.AppSettings["EmailTo"].ToString();

#endif

message.Body = body2 + body;

message.IsBodyHtml = true;

log.Info(message.Body);

string[] to = sender.Split(new char[] { ';' }, StringSplitOptions.RemoveEmptyEntries);

foreach (var address in to)

{

if (!string.IsNullOrEmpty(address))

{

if (address.IndexOf("@") < 1)

message.To.Add(new MailAddress(address + "@concentrix.com"));

else

message.To.Add(new MailAddress(address));

}

}

// ATTACHMENTS

if (File != null)

{

for (int i = 0; i < File.Length; i++)

{

string archivo = File[i];

if (archivo != null && archivo != "")

{

Attachment attach3 = new Attachment(archivo);

message.Attachments.Add(attach3);

}

}

}

log.Info("Attachemnts done");

client.Send(message);

}

catch (SmtpException smtpex)

{

//log.Info("smtp error " + smtpex.ToString());

throw smtpex;

}

catch (Exception ex)

{

log.Info(ex.ToString());

throw ex;

}

finally

{

if (message != null)

message.Dispose();

if (client != null)

client.Dispose();

}

}