using System;

using System.Collections.Concurrent;

using System.Reflection;

using Microsoft.Exchange.WebServices.Data;

using System.Net;

using System.Text;

using System.IO;

To include

Install-Package Microsoft.Exchange.WebServices

class Program

{

static BlockingCollection<string> queue = new BlockingCollection<string>();

static ExchangeService service;

static string[] File = new string[] { "","","","",""};

string ExchangeServiceUrl = string.Empty;

static void Main(string[] args)

{

StartExchangeService();

// CARGUE CORREOS

GetEmails();

var folder = Folder.Bind(service, WellKnownFolderName.DeletedItems);

int conteo = queue.Count;

// SI NO HAY CORREOS SALGA

if (conteo == 0)

{

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

DataAccess.DA\_Dashboard(queue.Count);

Environment.Exit(0);

}

// RECORRA CADA EMAIL

foreach (var item in queue.GetConsumingEnumerable())

{

ProcessEmail(item);

// LUEGO DE PROCESAR PASELO A DELETE FOLDER

Item item2 = Item.Bind(service, item.ToString());

item2.Move(folder.Id);

**// SI NO MAS CORREOS SALGA, DE LO CONTRARIO QUEDA TRABADO**

if (--conteo == 0) break;

}

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

Environment.Exit(0);

}

static private void ProcessEmail(string itemId)

{

Item item = null;

StringBuilder sb = new StringBuilder();

string EmailSubject = "";

string EmailBody = " ";

string body = "";

string body2 = "";

string sender = "";

string ending = "<br>Please review the employee information and resend it for processing.<br><br>Regards,<br>CNX HR Auto Response<br><br>";

try

{

item = Item.Bind(service, itemId);

if (item is EmailMessage)

{

EmailMessage email = item as EmailMessage;

//if (string.IsNullOrEmpty(email.Subject))

//{

// DataAccess.SendEmail("Error EmplVerif", "Missing subject", "");

// return;

//}

Email e = new Email();

sender = email.Sender.Address;

string sendername = email.Sender.Name;

string dtSended = email.DateTimeSent.ToString().Trim();

string subject = email.Subject.ToUpper();

string emailCreated = email.DateTimeCreated.ToString();

EmailBody = "<br>Hi " + sendername + "<br><br>";

body = email.Body.Text;

int pos1 = body.IndexOf("Dear ");

string first = "";

string last = "";

string empID = "";

if (pos1>0)

{

string msg = body.Substring(pos1 + 5);

pos1 = msg.IndexOf(",");

last = msg.Substring(0, pos1);

msg = msg.Substring(pos1+1);

pos1 = msg.IndexOf(",");

first = msg.Substring(0, pos1);

}

**getAttachments(email);**

if (File[0] == "")

{

EmailBody = EmailBody + "We did not find any attachment or consent form in your request. Thus, we are unable to process it.<br><br>" + body2 + ending;

DataAccess.SendEmail(EmailSubject, EmailBody, sender);

return;

}

byte[] doc1 = GetFile(File[0]);

byte[] doc2 = GetFile(File[1]);

byte[] doc3 = GetFile(File[2]);

byte[] doc4 = GetFile(File[3]);

byte[] doc5 = GetFile(File[4]);

string[] ext = new string[] { "", "", "", "", "" };

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

{

int pos = File[i].IndexOf(".");

if (pos > 0) ext[i] = File[i].Substring(pos + 1);

}

DataAccess.InsertDB(empID, first, last, doc1, ext[0], doc2, ext[1]);

}

}

catch (Exception es)

{

string err1 = es.ToString();

log.Error(err1);

DataAccess.SendEmail("Employee Last Pay Error ", err1, "");

}

}

**// CONVIERTALO PARA CARGARLO A BASE DE DATOS**

static private byte[] GetFile(string File)

{

if (File == "") return null;

FileStream stream = new FileStream(File, FileMode.Open, FileAccess.Read);

BinaryReader reader = new BinaryReader(stream);

byte[] doc = reader.ReadBytes((int)stream.Length);

reader.Close();

stream.Close();

return doc;

}

**static private void getAttachments(EmailMessage email)**

{

string filename = "";

int cual = 0;

string path = "C:\\temp\\FinalPay\\";

FileAttachment fileAttach = null;

int count = email.Attachments.Count;

for (int i = 0; i < count; i++)

{

if (cual == File.Length) break;

int size1 = email.Attachments[i].Size;

if (size1 > 10000)

{

fileAttach = email.Attachments[i] as FileAttachment;

filename = path + fileAttach.Name;

File[cual++] = filename;

try

{

System.IO.File.Delete(filename);

}

catch (Exception) { }

FileStream theStream = new FileStream(filename, FileMode.OpenOrCreate,

FileAccess.ReadWrite);

fileAttach.Load(theStream);

theStream.Dispose();

}

}

}

static private string Rename(string file)

{

int pos = file.IndexOf(".");

string newFile= file.Substring(0, pos) + "-2" + file.Substring(pos);

System.IO.File.Move(file, newFile);

return newFile;

}

static private void StartExchangeService()

{

try

{

service = new ExchangeService(ExchangeVersion.Exchange2010\_SP2);

Uri uri = new Uri("https://outlook.office365.com/EWS/Exchange.asmx");

IWebProxy proxy = WebRequest.GetSystemWebProxy();

bool isBypassed = proxy.IsBypassed(uri);

if (!isBypassed)

{

//if proxy is needed, then provide network credentials

proxy.Credentials = CredentialCache.DefaultNetworkCredentials;

service.WebProxy = proxy;

}

//https://msdn.microsoft.com/en-us/library/office/dn567668(v=exchg.150).aspx

#if DEBUG

service.Credentials = new WebCredentials("tdal6392@sa.convergys.com", "pvnptczcvqqwmlxy", "sa.convergys.com"); //enter your credentials to test // service.Credentials = new WebCredentials("tdal6392@sa.convergys.com", "pvnptczcvqqwmlxy", "sa.convergys.com"); //enter your credentials to test

#else

service.Credentials = new WebCredentials("hr.autoresponse@concentrix.com", "CNX#HR%rp@", "concentrix.com"); //enter your credentials to test

// service.Credentials = new WebCredentials("notification.robot@na.convergys.com", "wjkxyfrdxgqpzxjy", "na.convergys.com");

#endif

service.Url = uri;

log.Info("Service.Credentials");

}

catch (Exception ex)

{

string msgErr = "StartExchangeService \\n\\n " + ex.ToString();

DataAccess.SendEmail("Employee Verificiation - Error ONStart", msgErr, "");

log.Error(msgErr);

}

}

static private void GetEmails()

{

log.Info("Getting pending emails.");

try

{

bool process = true;

if (process)

{

int offset = 0;

int pageSize = 50;

bool more = false;

ItemView view = new ItemView(pageSize, offset, OffsetBasePoint.Beginning);

view.OrderBy.Add(ItemSchema.DateTimeReceived, SortDirection.Descending);

view.PropertySet = PropertySet.IdOnly;

FindItemsResults<Item> results;

do

{

results = service.FindItems(WellKnownFolderName.Inbox, view);

foreach (var item in results.Items)

{

queue.Add(item.Id.UniqueId);

}

more = results.MoreAvailable;

if (more)

{

view.Offset += pageSize;

}

} while (more);

}

}

catch (Exception ex)

{

string msgErr = "GetPendingEmails\\n\\n " + ex.ToString();

//ReportError(msgErr);

}

}

}

Para cargarlo a base de datos

static public bool InsertDB(string EmpID, string first, string last, byte[] doc1, string ext1 )

{

string sql = "INSERT into HR\_FinalPay ";

sql = sql + " (Created, EmpID, LastName, FirstName, Doc1, Ext1) ";

sql = sql + " VALUES ( @Created, @EmpID, @last, @first, @doc1, @ext1) ";

try

{

string SqlConn = ConfigurationManager.ConnectionStrings["SqlTest"].ConnectionString;

SqlConnection sqlcon = new SqlConnection(SqlConn);

sqlcon.Open();

using (SqlCommand comm = new SqlCommand(sql, sqlcon))

{

comm.Parameters.AddWithValue("@EmpID", EmpID);

comm.Parameters.AddWithValue("@last", last);

comm.Parameters.AddWithValue("@first", first);

if (ext1!="")

comm.Parameters.Add("@doc1", SqlDbType.Image, doc1.Length).Value = doc1;

else

comm.Parameters.Add("@doc1", SqlDbType.Image, doc1.Length).Value = DBNull.Value;

comm.ExecuteNonQuery();

}

sqlcon.Close();

sqlcon.Dispose();

return true;

}

catch (Exception ws)

{

string msgErr = "Error when INSERTING HR\_FinalPay " + sql + " " + ws.ToString();

SendEmail("Error Emp Verification Insert ", msgErr, "");

}

return false;

}

1. Go to <https://portal.office.com/account/>
2. tdal6392@sa.convergys.com

2. Go to Security & privacy

3. Go to Additional security verification

 Or

4. Go to Create and manage app passwords

