**Class Generic Email**

**public** **void** Send\_Email()

{

System.Net.Mail.MailMessage mailMessage;

System.Net.Mail.SmtpClient myMail;

System.Net.Mail.MailAddress mailFrom;

System.Net.Mail.MailAddress mailTo;

System.Net.Mail.Attachment attachment;

System.Net.Mail.AttachmentCollection attachementCollection;

System.Net.Mail.MailAddress bccmailAddress;

System.Net.Mail.MailAddressCollection BCC\_Collection;

System.Net.Mail.MailAddress ccmailAddress;

System.Net.Mail.MailAddressCollection CC\_Collection;

System.Net.Mail.MailAddressCollection To\_Collection;

System.Exception excObj;

System.Net.Mail.MailAddress bcc ;

System.Net.Mail.MailAddress cc;

SysEmailContents mailBody;

SysEmailTable table;

SysEmailMessageTable message;

Map mappings;

FileName fileName;

userinfo userInfo;

SysEmailParameters parameters;

**int** loop\_var;*//use*

**str** mail;*//c*

**str** receiverMailAddress;

**str** smtpServer;

**str** mailSubject;

**str** ccMail;

**str** pwd;

**str** bccMail;

**int** SMTPPort;

**anytype** tempAny;

*// taking sending ID from Template we made we are giving Template ID here*

table = SysEmailTable::find("Sohaib36");*//emailTemplateID*

message =SysEmailMessageTable::find("Sohaib36", table.DefaultLanguage);

*// setting attachment File path*

fileName = @'C:\Users\internsohaib\Desktop\Email.txt';

*// taking parameters from Email Parameters*

parameters = SysEmailParameters::find();

**new** InteropPermission(InteropKind::ClrInterop).assert();

*// mapping values on Template*

mappings = **new** Map(Types::String,Types::String);

mappings.insert("CustomerName", "Sohaib"); *// this will replace variable with actual value.*

mappings.insert("salesID", "123");

mappings.insert("CompanyName", "Sagacious");

mailBody = SysEmailMessage::stringExpand(message.Mail,mappings);

mailSubject = SysEmailMessage::stringExpand(message.Subject,mappings);

**try**

{

smtpServer = SysEmaiLParameters::find(**false**).SMTPRelayServerName;*// using the SMTP server ip //setup in email Parameters*

mailMessage= **new** System.Net.Mail.MailMessage();

*// Setting Sender*

**if** (This\_sender=="")

{

This\_sender=parameters.SMTPUserName;

*//mailFrom = new System.Net.Mail.MailAddress(This\_sender);*

mailMessage.set\_From( **new** System.Net.Mail.MailAddress(This\_sender));

info(**strFmt**(" Defualt Sender Address is Set %1",This\_sender));

}

**else**

{

*//mailFrom=new System.Net.Mail.MailAddress(This\_sender);*

mailMessage.set\_From( **new** System.Net.Mail.MailAddress(This\_sender));

info(**strFmt**(" User Sender Address is Set %1",This\_sender));

}

mailmessage.set\_Subject(mailSubject);

mailmessage.set\_Body(mailBody);

mailMessage.set\_IsBodyHtml(**true**);

*//\*\*\*\*\*\*\*\*\*CC Added \*\*\*\*\*\*\*\*\*\**

CC\_Collection=**new** System.Net.Mail.MailAddressCollection();

**if** (**conLen**(This\_cc)>**0**)

{

CC\_Collection =mailMessage.get\_CC();*// taking refrence of BCC cotainer*

**for**(loop\_var=**1**; loop\_var<=**conlen**(This\_cc);loop\_var++ )

{

ccMail=**conpeek**(This\_cc,loop\_var);

cc=**new** System.Net.Mail.MailAddress(ccMail);

CC\_Collection.Add(cc);

}

*// adding mail to container*

info("CC added");

}*// ending IF Block*

**else**

{

info("No one Is Added to CC");

}*// CC Else End Block*

*//attaching Attachment*

attachementCollection = mailMessage.get\_Attachments();

attachment = **new** System.Net.Mail.Attachment(fileName);

attachementCollection.Add(attachment);

*// setting up SMTP protocol for Mail*

SMTPPort = SysEmaiLParameters::find(**false**).SMTPPortNumber;

myMail = **new** System.Net.Mail.SmtpClient(smtpServer, SMTPPort);

*//SSL Use for Secure Mail Because SMTP use secure connection TCP/IP*

myMail.set\_EnableSsl(**true**); *// For SSL enabled mail servers. Ex: gmail, smtp.gmail.com, port 465 or 587*

pwd = SysEmaiLParameters::password();

mymail.set\_Credentials(**New** System.Net.NetworkCredential(parameters.SMTPUserName, pwd));

**if** (**conLen**(This\_bcc)>**0**)

{

BCC\_Collection =mailMessage.get\_Bcc();*// taking refrence of BCC cotainer*

**for**(loop\_var=**1**; loop\_var<=**conlen**(This\_bcc);loop\_var++ )

{

bccMail=**conpeek**(This\_bcc,loop\_var);

bcc=**new** System.Net.Mail.MailAddress(bccMail);

BCC\_Collection.Add(bcc);

}

*// adding mail to container*

info("BCC added");

}*// ending IF Block*

**else**

{

info("No one Is Added to BCC");

}*// BCC Else End Block*

*// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Setting Receiver \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**if** (**conlen**(This\_receivers)==**0**)

{

info("please Set Receiver First (Set Receiver method)");

}

**else**

{

To\_Collection=mailMessage.get\_To();

**for**(loop\_var=**1**; loop\_var<=**conlen**(This\_receivers);loop\_var++ )

{

receiverMailAddress=**conpeek**(This\_receivers, loop\_var);

mailTo= **new** System.Net.Mail.MailAddress(receiverMailAddress); *// See if Memory Leakage*

To\_Collection.Add(mailTo);

}*// for loop end bracket*

info("Recepient added");

}*// else end bracket*

info(This\_sender);

mailMessage.set\_From( **new** System.Net.Mail.MailAddress(This\_sender));

mymail.Send(mailmessage);

info("Mail Sent");

info("All mails Sent");

}*// try end bracket*

**catch**

{

excObj = CLRInterop::getLastException();

**if**(excObj)

{

error(excObj.ToString());

}

}

mailMessage.Dispose();

CodeAccessPermission::revertAssert();

}

**class** Email\_Generic\_Class

{

**container** This\_cc;

**container** This\_receivers;

**container** This\_bcc;

**str** This\_sender;

}

**public** **void** Set\_BCC\_Through\_Container(**container** BCC)

{

**int** Number\_of\_BCC;

this\_bcc=BCC;

Number\_of\_BCC=**conLen**(this\_bcc);

info(**strFmt**("Total BCC =%1",Number\_of\_BCC));

}

**public** **void** set\_CC\_Through\_container(**container** cc)

{

This\_cc=cc;

info(**strFmt**("Total CC Added =%1",**conLen**(This\_cc)));

}

**public** **void** Set\_Receiver\_through\_Container(**container** params)

{

This\_receivers=params;

info(**strFmt**("Total Receivers =%1",**conLen**(This\_receivers)));

}

**public** **void** set\_Sender(**str** sender)

{

This\_sender=sender;

}

Testing JOB

**static** **void** Testing\_Email()

{

*// Class*

Email\_Generic\_Class e= **new** Email\_Generic\_Class();

**container** Receiver,BCC,CC;

*//tables*

Sohaib\_Email emails;

*// setting Receiver*

**str** readitmIds;

**int** conLength , cnt;

**while** **select** Emails **from** emails

**where** emails.Designation == Sohaib\_Email\_Enum::IT

{

Receiver += [emails.Emails];

}

e.Set\_Receiver\_through\_Container(Receiver);

*//Setting BCC*

**while** **select** Emails **from** emails

**where** emails.Designation == Sohaib\_Email\_Enum::CEO

{

BCC += [emails.Emails];

}

*//Setting CC*

**while** **select** Emails **from** emails

**where** emails.Designation == Sohaib\_Email\_Enum::HR

{

CC += [emails.Emails];

}

e.Set\_BCC\_Through\_Container(BCC);

e.set\_CC\_Through\_container(CC);

e.set\_Sender("sohaibanwaar36@gmail.com");

e.Send\_Email();

}