**Generic Email :**

**static** **void** Send\_Email(**container** receivers)

{

System.Net.Mail.MailMessage mailMessage;

System.Net.Mail.SmtpClient myMail;

System.Net.Mail.MailAddressCollection mailcoll;

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;

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

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

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

**int** loop\_var;

**str** mail;

**str** receiverMailAddress;

SysEmailContents mailBody;

**str** smtpServer;

**str** mailSubject;

**str** ccMail;

**str** bccMail;

**int** SMTPPort;

System.Exception excObj;

System.Net.Mail.MailAddress bcc ;

System.Net.Mail.MailAddress cc;

**anytype** tempAny;

Map mappings;

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

userinfo userInfo;

**str** pwd;

SysEmailParameters parameters = SysEmailParameters::find();

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

receiverMailAddress = "sohaibanwaar1203@gmail.com";

bccMail = "15009065388@umt.edu.pk";

ccMail="15026020324@umt.edu.pk";

*// 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");

mailFrom = **new** System.Net.Mail.MailAddress(parameters.SMTPUserName,"Name");

mailTo = **new** System.Net.Mail.MailAddress(receiverMailAddress);

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

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

mailcoll = **new** System.Net.Mail.MailAddressCollection();

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(mailFrom,mailTo);

mailmessage.set\_Subject(mailSubject);

mailmessage.set\_Body(mailBody);

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

*// attaching 1 BCC*

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

BCC\_Collection.Add(bccMail);*// adding mail to container*

*//attaching 1 CC*

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

CC\_Collection.Add(ccMail);*// adding mail to container*

*//attaching Attachment*

attachementCollection = mailMessage.get\_Attachments();

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

attachementCollection.Add(attachment);

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**(receivers)==**0**)

{

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

}

**else**

{

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

{

receiverMailAddress=**conpeek**(receivers, loop\_var);

info(receiverMailAddress);

mailTo= **new** System.Net.Mail.MailAddress(receiverMailAddress);

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

mymail.Send(mailmessage);

}

}*// else end bracket*

info("All mails Sent");

}*// try end bracket*

**catch**

{

excObj = CLRInterop::getLastException();

**if**(excObj)

{

error(excObj.ToString());

}

}

mailMessage.Dispose();

CodeAccessPermission::revertAssert();

}

**Testing Job:**

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

{

*// Class*

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

**container** param;

*//tables*

Sohaib\_Email emails;

**str** readitmIds;

**int** conLength , cnt;

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

{

param += [emails.Emails];

}

e.Set\_Receiver(param);

Email\_Generic\_Class::Send\_Email(param);

}