Code Walkthrough (Distributor)

The distributor is going to check for new notifications in the V\_NOTIFICATION\_INFORMATION view.

According with our example it will read the next information:

|  |  |
| --- | --- |
| **V\_NOTIFICATION\_INFORMATION** | |
| **Column Name** | **Data** |
| NOTF\_ID | 1 |
| NOTF\_INFORMATION | <xmldata>  <data VESG\_THEMA = “ Mathematik” />  <data VESG\_VERANSTALTUNGORT = “Erfurt”/>  </xmldata > |
| SBPN\_ID | 9 |
| SBER\_ID | 1 |
| NFEV\_ID | 1 |
| NFEV\_DIRECTINFORMATION | null |
| NFEV\_EMAILCC | event@provider.com |
| NFEV\_EMAILBCC | event2@provider.com |
| NFEV\_ATTACHMENT | <attachments><attachment id=”1”/></attachments> |
| DEVC\_ID | 1 |
| DEVC\_ADDRESS | user@email.com |
| SBTP\_CODE | 6 |
| SBTP\_DECODE | New Courses |
| SBTP\_EMAILADR | events@institution.com |
| SBTP\_EMAILSUB | <xmldata>  <fields><field name=“ VESG\_BEGINN”/> <field name=“ VESG\_ENDE”/></fields>  <data subject=“ New Course #VESG\_BEGINN # To #VESG\_ENDE#” />  </xmldata> |
| SBTP\_EMAILREPLYADR | events2@institution.com |
| SBTP\_EMAILCC | events3@institution.com |
| SBTP\_EMAILBCC | events4@institution.com |
| SBTP\_ISDYNAMICSUB | 1 |
| NOTX\_TEXT | <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  <xsl:template match="/">  <xsl:for-each select="xmldata/data">  Betreff: Anbieterprofil geändert  Werte Mitarbeiter der Akkreditierungsstelle,  die Daten des akkreditierten Anbieters  <xsl:value-of select="@ABIE\_NUMMER" />  wurden vom Anbieter am <xsl:value-of select="@ABIE\_\_x005F\_x0023\_PTS" />  geändert und sind zu prüfen.  </xsl:for-each>  </xsl:template>  </xsl:stylesheet> |
| CMMD\_CODE | 1 |
| CMMD\_DECODE | Email |

*ListNotifications = Helpers.bobjNotifications.Select ();*

The Distributor will take this information and pass it into a list so we can work with it.

*While(read ListNotifications)*

*{*

*if (notificationEventDirectInformation != null)*

*{*

*Ni = OpenInformation(notificationEventDirectInformation);*

*}*

Checks if the notification is direct (without subscription) and if it is then calls the method OpenInformation to open the xml information that the NFEV\_DIRECTINFORMATION field contains. This method passes everything into an object.

*MessageBody=XslTransform(ListNotifications*);

Now that the distributor knows the name of the xslt file, it will add the format to the notification information.

*If ListNotifications.communicationmethod=email*

*{*

*SendEmail(NotificationInformationList, MessageBody);*

*}*

*Else*

*{*

*SendSMS(NotificationInformationList, MessageBody);*

*}*

When the information is ready, the distributor will check the communication method, and call the right method to send it.

*SendEmail()*

*{*

*msg.To.Add(new MailAddress(deviceAddress));*

*if (subscriptionTypeEmailCc != null)*

*msg.CC.Add(new MailAddress(subscriptionTypeEmailCc));*

*if (notificationEventEmailCc != null)*

*msg.CC.Add(new MailAddress(notificationEventEmailCc));*

*if (subscriptionTypeEmailBcc != null)*

*msg.Bcc.Add(new MailAddress(subscriptionTypeEmailBcc));*

*if (notificationEventEmailBcc != null)*

*msg.Bcc.Add(new MailAddress(notificationEventEmailBcc));*

*msg.From = new MailAddress(subscriptionTypeEmailAddress);*

*if (subscriptionTypeReplyAddress != null)*

*msg.ReplyTo = new MailAddress(subscriptionTypeReplyAddress);*

Adds the correct addresses for sending the email

*msg.Subject = EmailSubject(NotificationInformation);*

Calls the EmailSubject method to make the replacement in case the subject is dynamic

*if (Ni.notificationEventAttachment != null)*

*{*

*IList Attachments = new List<bobjAttachments>();*

*Attachments = GetAttachments(notificationEventAttachment);*

Check if the notification has attachments, if it has then calls the GetAttachments method to open the xml field and get the attachments from the database.

*foreach (attachment in Attachments)*

*{*

*MemoryStream ms = new MemoryStream(anlaDatei);*

*Attachment attached = new Attachment(ms, attach.anlaName, MediaTypeNames.Application.Octet);*

*msg.Attachments.Add(attached);*

*}*

*}*

Adds the attachment into the email.

*SmtpClient clienteSmtp = new SmtpClient(Helpers.SystemEinstellungen().SelectServer());*

*clienteSmtp.Credentials = new System.Net.NetworkCredential(Helpers.SystemEinstellungen().SelectUser(),*

*Helpers.SystemEinstellungen().SelectPassw());*

Gets the server and credentials from the database*.*

*try*

*{*

*clienteSmtp.Send(msg);*

Sends the email.

*}*

*catch (Exception ex)*

*{*

*ServiceLogging.LogError(ex);*

*}*

*}*

*EmailSubject()*

*{*

*XmlDocument xDoc = new XmlDocument();*

*xDoc.LoadXml(Ni.subscriptionTypeEmailSubject);*

*XmlNodeList XmlData = xDoc.GetElementsByTagName("xmldata");*

*XmlNodeList Data = ((XmlElement)XmlData[0]).GetElementsByTagName("data");*

*string Subject=Data[0].Attributes[0].Value;*

*if (subscriptionTypeIsDynamicSub == "1")*

*{*

*XmlNodeList Fields = ((XmlElement)XmlData[0]).GetElementsByTagName("fields");*

*XmlNodeList Field = ((XmlElement)XmlData[0]).GetElementsByTagName("field");*

*foreach(XmlNode node in Field)*

*{*

*Subject=Subject.Replace("#"+node.Attributes[0].Value+"#",*

*FindField(notificationInformation, node.Attributes[0].Value));*

*}*

*}*

*return Subject;*

*}*

The Emailsubject method checks if the email subject is dynamic, if it is, then opens the xml field and calls the FindField method to find the correct fields to replace in the subject.