Aufgabe: Mails inklusive Anlagen verarbeiten:

Varianten:

1. Mail mit reinem Hinweis auf Anlage
2. Mail mit Inhalt und (z.B. erläuternden) Anlagen

Verarbeitung:

Zu 1: Es reicht, die Anlagen zu speichern. Mail kann gelöscht werden.

Zu 2:

2a. Mail komplett mit Anlagen speichern

2b. Mail und Anlagen getrennt speichern

2.ba) Hinweis auf Anlagen in der Mail

2.bb) Kein Hinweis auf Anlagen

Ansätze für Outlook (msg) und offene Systeme (eml)

|  |  |  |  |
| --- | --- | --- | --- |
| **Schritt** | **Msg** | **Eml** | **Bemerkung** |
| 1. Mail mit reinem Hinweis auf Anlage | Bisher nur durch manuelles Verschieben der Anlage in ScanInput-Ordner | In Mail-Teil vom Start-Form enthalten | Es reicht, die Anlagen zu speichern. Mail kann gelöscht werden. |
| 2. Mail mit Inhalt und (z.B. erläuternden) Anlagen |  |  |  |
| 2a. Mail komplett mit Anlagen speichern | In FolderInput unter MsgHandle | In FolderInput und emlHandle | Beides schon enthalten |
| 2.ba) Hinweis auf Anlagen in der Mail |  |  | D.h. Mail und Anlagen getrennt erfassen und speichern |
| 2.bb) Kein Hinweis auf Anlagen |  |  |  |

Verwaltung der Anlagen ist für Msg und Eml nicht gemeinsam möglich, da verschiedene Algorithmen! Msg über Namespace Microsoft.Outlook und Eml über Namespace OpenPop.

Erforderlich daher zwei getrennte Routinen (private sub’s) im Form FolderInput (im Sub msgHandle und emlHandle)

Eml Vorhanden:

Public Sub EmlHandle(ByVal MyEml As String)

Dim file As FileInfo = New FileInfo(MyEml)

Dim loadedMsg As Message = Message.Load(file)

Dim msgBody As String = ""

'WebBrowser1.Navigate("")

Dim body As MessagePart = loadedMsg.FindFirstHtmlVersion()

If body IsNot Nothing Then

'FileContent = body.GetBodyAsText()

msgBody = body.GetBodyAsText()

' MsgBox(msgBody)

Else

body = loadedMsg.FindFirstPlainTextVersion()

If body IsNot Nothing Then

'FileContent = body.GetBodyAsText()

msgBody = body.GetBodyAsText()

' MsgBox(msgBody)

End If

End If

DokumentTextBox.Text = "EMail von " & loadedMsg.Headers.From.DisplayName.ToString

BetreffTextBox.Text = loadedMsg.Headers.Subject

DokDatumTextBox.Text = loadedMsg.Headers.DateSent.ToShortDateString

If msgBody.Length > 2000 Then

KommentarTextBox.Text = "Text zu lang, bitte ggf. OCR-Feld prüfen!"

Else

KommentarTextBox.Text = tempOCRString

End If

'Behandlung der Anlagen: Laden in Liste selAttachment und Anzeigen in Textbox

selAttachments.Clear()

If loadedMsg.FindAllAttachments.Count > 0 Then

selAttachments = loadedMsg.FindAllAttachments()

AnlagenTextBox.Text = loadedMsg.FindAllAttachments.Count.ToString

For Each anl In loadedMsg.FindAllAttachments

Hier Anlagen zwischenspeichern:

*For Each anl in loadedMsg.FinAllAttachments*

*WriteToFile(anl.Body, tempPath & "temp\_" & anl.FileName)*

*Mit:*

*Private Function WriteToFile(ByVal Source As Byte(), ByVal Filename As String) As Boolean*

*Try*

*My.Computer.FileSystem.WriteAllBytes(Filename, Source, False)*

*If IO.File.Exists(Filename) Then*

*Return True*

*Else : Return False*

*End If*

*Catch ex As IO.IOException*

*MessageBox.Show(ex.Message.ToString(), "IOException")*

*Catch ex As Exception*

*MessageBox.Show(ex.Message.ToString(), "Exception")*

*End Try*

*Return False*

*End Function*

lboxSaveAtt.Items.Add(anl.FileName)

AnlagenIDTextBox.Text += loadedMsg.Headers.MessageId & ", "

Next

AnlagenIDTextBox.Text = AnlagenIDTextBox.Text.Substring(0, AnlagenIDTextBox.Text.Length - 2)

End If

'MsgBox(webMsg)

'WebBrowser1.DocumentText = MyString 'loadedMsg.MessagePart.GetBodyAsText

WebBrowser1.Document.Body.InnerHtml = msgBody 'WebBrowser1.DocumentText = msgBody.ToString

tempOCRString = WebBrowser1.Document.Body.InnerText

RichTextBoxOCR.Text = tempOCRString

InhaltOCRTextBox.Text = tempOCRString

End Sub

Msg Vorhanden:

Private Sub MsgHandle(ByVal Msg As String)

Dim oApp As New outlook.Application

Dim OMItem As outlook.MailItem

Try

OMItem = CType(oApp.CreateItemFromTemplate(Msg), outlook.MailItem)

If OMItem.BodyFormat = outlook.OlBodyFormat.olFormatPlain Then

WebBrowser1.DocumentText = OMItem.Body

Else

'WebBrowser1.DocumentText = OMItem.HTMLBody

WebBrowser1.Document.Body.InnerHtml = OMItem.HTMLBody

End If

Dim body As String = WebBrowser1.Document.Body.InnerText

' MsgBox(body)

FileContent = "Gesendet: " & OMItem.SentOn & vbCrLf & "Von: " & OMItem.SenderName & vbCrLf & "Betreff: " & OMItem.Subject & vbCrLf & body

tempOCRString = FileContent

' WebBrowser1.DocumentText = tempOCRString

DokumentTextBox.Text = "EMail von " & OMItem.SenderName.ToString

BetreffTextBox.Text = OMItem.Subject

DokDatumTextBox.Text = OMItem.SentOn.ToShortDateString

If OMItem.Attachments.Count > 0 Then AnlagenTextBox.Text = OMItem.Attachments.Count.ToString

'Test

selAttachments.Clear()

If OMItem.Attachments.Count > 0 Then

Hier Anlagen zwischenspeichern:

*attName = tempPath* *& "temp\_" & "-" & locGDL(i).gdlDokName*

*For Each anl As outlook.Attachment In OMItem.Attachments*

*'Anlage temporär speichern*

*Try*

*anl.SaveAsFile(tempPath & "temp\_" & anl.FileName)*

*Catch ex As Exception*

*MsgBox(ex.Message)*

*End Try*

lboxSaveAtt.Items.Add(anl.FileName)

AnlagenIDTextBox.Text += OMItem.EntryID & ", " '.MessageId & ", "

Next

AnlagenIDTextBox.Text = AnlagenIDTextBox.Text.Substring(0, AnlagenIDTextBox.Text.Length - 2)

End If

'end TEst

If OMItem.Body.Length > 2000 Then

KommentarTextBox.Text = "Text zu lang, bitte ggf. OCR-Feld prüfen!"

Else

KommentarTextBox.Text = OMItem.Body

End If

RichTextBoxOCR.Text = tempOCRString

InhaltOCRTextBox.Text = tempOCRString

Catch ex As Exception

WebBrowser1.Navigate("about: blank")

MsgBox(ex.Message)

Finally

oApp = Nothing

OMItem = Nothing

End Try

End Sub

Nach Auswahl in AnlagenListbox (Eml):

Private Sub LboxSaveAtt\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles lboxSaveAtt.SelectedIndexChanged

Dim ix As Integer = lboxSaveAtt.SelectedIndex()

If ix < 0 Then

MsgBox("Bitte zunächst eine Anlage auswählen")

Exit Sub

End If

btSave.Enabled = True

Dim AttName As String = selAttachments(ix).FileName

isAttachment = True

''speichern der Mail in Liste

FillDokInhalt()

ClearText()

If DokInhalt.Count > 0 Then

For Each el In DokInhalt

If el.docID.ToString.Contains(anzahlDS + 1) Then

'MsgBox(el.betreff)

FdBearbeitungsVermerkTextBox.Text = "Anlage " & ix + 1 & " zu " & el.dokName

End If

Next

End If

'Vorausfüllen der Textfelder:

Dim temp As String = DokumentTextBox.Text

IstAnlageCheckBox.Checked = True

' MsgBox(selAttachments(ix).Body.ToString)

If WriteToFile(selAttachments(ix).Body, tempPath & AttName) Then

' Process.Start(tempPath & AttName)

FileInhalt(tempPath & AttName)

End If

End Sub

selAttachments ist Liste of MessageParts -> OpenPop!

Msg (Outlook) kann nicht auf diese Liste zugreifen bzw. in ihr speichern!

* Zwischenspreichern der Anlagen bei Abfrage nach Anlagen

OK, durch Änderungen (Kasten) werden die Anlagen jetzt zwischengespeichert.

Anzeige in Listbox auch ok.

Weiter:?

Übernahme als neues Dokument dürfte ohne Probleme möglich sein.

Aber damit ist keine Verknüpfung mit Dokument gegeben ->

**Variante 1:**

In der Mail wie bisher Anlagenanzahl anzeigen und in den Anlagen Hinweise auf Ursprungsmail einfügen.

Vorteil: relativ leicht umzusetzen

Nachteil: In Mail nur allgemeiner Hinweis auf Anlagen, keine Verknüpfung!

**Variante 2:**

In der DB ist noch die Tabelle Anlagen vorhanden. Nutzen??

Würde aber das gesamte Konzept ändern -> bei jeder Detailanzeige der Tabelle Dokumente müsste eine Listbox mit Anlagen angezeigt werden, aus denen dann auch das Originaldokument aufgerufen werden müsste.

Betrifft: DocEdit, DocListe, …

Umsetzung:

1. Mail verarbeiten -> ok
2. Anlagen werden zwischengespeichert -> ok
3. Anzeige der Anlagen erfolgt in ListBox -> ok
4. Auswahl in ListBox führt zu neuer Verarbeitung mit neuer DokID und neuem DokName
5. Dazu sind DokID der AusgangsMail und eine AnlagenID erforderlich -> Refresh

③ Private Sub lboxSaveAtt\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles lboxSaveAtt.SelectedIndexChanged

‘refresh -> DokID der Mail ermitteln

‘darf erst erfolgen, wenn Dokument(Mail) gespeichert wurde.

Validate()

DokumenteBindingSource.EndEdit()

DokumenteTableAdapter.Update(\_FreeDMS\_StartDBDataSet.Dokumente)

currDokID = CountData(CurrDB)

MsgBox(currDokID.ToString)

MsgBox(lboxSaveAtt.SelectedItem.ToString)

isAttachment = True

FileInhalt(tempPath & "temp\_" & lboxSaveAtt.SelectedItem.ToString)

‘jetzt in Tabelle Anlagen speichern -> Fehler: FileInhalt ist noch nicht ausgefüllt

Gelöst:

Dim anyRow = \_FreeDMS\_StartDBDataSet.Anlagen.NewAnlagenRow

anyRow.DokID = currDokID

anyRow.AnlagenID = CountData(CurrDB).ToString -> ist der Aktuelle Datensatz

FreeDMS\_StartDBDataSet.Anlagen.AddAnlagenRow(anyRow)

End Sub

Gehört nach SaveDoc():

…

…

'Wenn Attachment, in Tb Anlagen speichern

If isAttachment = True Then

Dim anyRow = \_FreeDMS\_StartDBDataSet.Anlagen.NewAnlagenRow

anyRow.DokID = lastID

anyRow.AnlagenID = (MaxID(CurrDB) + 1).ToString

\_FreeDMS\_StartDBDataSet.Anlagen.AddAnlagenRow(anyRow)

isAttachment = False

End If

'InhaltOCRTextBox.Text = tempOCRString

Try

Validate()

DokumenteBindingSource.EndEdit()

AnlagenBindingSource.EndEdit()

DokumenteTableAdapter.Update(\_FreeDMS\_StartDBDataSet.Dokumente)

AnlagenTableAdapter.Update(\_FreeDMS\_StartDBDataSet.Anlagen)

'erhöhen für dokNr.

Catch ex As Exception

Aufruf jetzt mit:

Private Sub lboxSaveAtt\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles lboxSaveAtt.SelectedIndexChanged

~~Validate()~~

~~DokumenteBindingSource.EndEdit()~~

~~DokumenteTableAdapter.Update(\_FreeDMS\_StartDBDataSet.Dokumente)~~

~~MailID = lastID + 1 ' currDokID = MaxID(CurrDB)~~

'MsgBox(currDokID.ToString & " - " & currDokName)

'MsgBox(lboxSaveAtt.SelectedItem.ToString)

isAttachment = True

BtSave.Enabled = True

FileInhalt(tempPath & "temp\_" & lboxSaveAtt.SelectedItem.ToString)

End Sub

Problem mit der DokID im gesamten FolderInput:

Die Zuordnung von

Dim lastID As Integer = CInt((FreeDMS\_Aux.FreeDMS.Aux.MaxID(CurrDB))) 'ID des letzten in der DB gespeicherten Datensatzes

Dim currDokID As Integer = lastID + 1 'ID des aktuellen Datensatzes (i.d.R lastID + 1)

Dim MailID As Integer 'DokID der Mail, wenn Anlagen gespeichert werden (für Tb. Anlagen)

Ist im Verlauf ist unklar. -> Prüfen!

Die ID des in Arbeit befindlichen Dokuments (CurrDokID) ist erforderlich

* Für den DokNamen (im Archiv)
* Für den Hinweis (Berabeitungsvermerk) bei weiterer Zuordnung (Zwischenspeicherung in DokInhalt)
* Für die Speicherung in der Tab. Anlagen wenn Anlagen verarbeitet werden

Die „virtuelle“CurrDokID wird aufbauend auf der lastID (wird beim FormLoad mit der Funktion MaxID(CurrDB) aus der aktuellen Tb Dokumente der geladenen DB ermittelt) hochgezählt mit jedem neuen Datensatz bis zum DB-Update. Sie ist erforderlich, da ansonsten jeder Datensatzeintrag in FolderInput in der Datenbank gespeichert werden müsste (damit wäre das System von DataTable/DataBinding obsolet).

Ablauf:

Beim FomLoad -> lastID ermittelt.

Die ID des neuen Datensatzes ist CurrDokID = lastID + 1.

Dieser Wert wird auf für DokName genutzt (Sub RenameDokFile):

Private Sub RenameDokFile(ByVal sDokName As String)

If mehrfachFlag = False Then

newFile = CbMandant.SelectedValue.ToString & "-" & CbAkte.SelectedValue.ToString & "-" & CbVorgang.SelectedValue.ToString & "#" & currDokID & System.IO.Path.GetExtension(sDokName)

DokNameTextBox.Text = newFile

lastDocName = newFile

Try

File.Copy(sDokName, ArchivOrdner & newFile, True)

Catch ex As Exception

MsgBox(ex.Message)

Exit Sub

End Try

WebBrowser1.Navigate("about:blank")

'und Zeit zum "Abschalten" lassen -> sonst Fehler: Datei noch geöffnet

System.Windows.Forms.Application.DoEvents()

System.Threading.Thread.Sleep(1000)

'und das Originaldokument löschen

Try

Kill(sDokName)

Catch ex As Exception

MsgBox(ex.Message)

End Try

MsgBox("Abgelegt unter Az: " & newFile)

End If

End Sub

Danach erfolgt Zwischenspeicherung (Sub FillDokInhalt).

Bisher wird hier CurrDokID + 1 gespeichert um beim Abruf, wenn mehrfachFlag=true,

die neue CurrDokID zu haben. Prüfen, ob erforderlich -> ist erforderlich

Bei btWeiterZuordnung\_Click wird CurrDokID (wurde in DokInhalt als docID übernommen) zu Prüfung genutzt:

Private Sub btWeiterZuordnung\_Click(sender As Object, e As EventArgs) Handles btWeiterZuordnung.Click

Dim dokName() As String

dokName = Split(lastDocName, "#")

mehrfachFlag = True

DokumenteBindingSource.AddNew()

For Each el In DokInhalt

If el.docID.ToString.Contains(CStr(currDokID)) Then

With el

DokumentTextBox.Text = .dokument

BetreffTextBox.Text = .betreff

BetragTextBox.Text = .betrag.ToString

AnlagenTextBox.Text = .anlagen

CbAbsender.SelectedValue = .absender 'AbsenderTextBox.Text = Str(.absender)

CbEmpfaenger.SelectedValue = .empfaenger

DokDatumTextBox.Text = .dokDatum.ToString

AufgenommenTextBox.Text = .aufgenommen

KommentarTextBox.Text = .kommentar

tempOCRString = .inhalt

InhaltOCRTextBox.Text = .inhalt

CbAblage.SelectedValue = .ablage 'AblageTextBox.Text = Str(.ablage)

CbTyp.SelectedValue = .typ 'TypTextBox.Text = Str(.typ)

CbStatus.SelectedValue = .status 'StatusTextBox.Text = Str(.status)

DokNameTextBox.Text = .dokName

If .istanlage Then IstAnlageCheckBox.CheckState = CheckState.Checked

AnlagenIDTextBox.Text = .anlagenid

If String.IsNullOrEmpty(el.bearbeit) Then

BearbVermerkTextBox.Text = " Siehe auch Az: " & dokName(0) & "/" & (el.docID - 1).ToString

Else

BearbVermerkTextBox.Text = el.bearbeit & vbCrLf & " Siehe auch Az: " & dokName(0) & "/" & (el.docID - 1).ToString

End If

End With

End If

Next

btWeiterZuordnung.Enabled = False

BtSave.Enabled = True

End Sub

Die Abfrage ist nicht mehr erforderlich, da DokInhalt jeweils nur 1x gefüllt wird!

-> Prüfen -> nein. ListOf() entfernt.

Danach erfolgt Speicherung und Erhöhung von CurrDokID um 1 (SaveDoc):

Private Sub SaveDoc()

Dim tempName As String = Nothing

'Abfrage ob die erforderlichern Felder gefüllt sind

.

.

'Wenn Dok mehrfach zugeordnet wird

'evt. vorhandene .ocr-Datei löschen

BtSave.Enabled = False

If mehrfachFlag = False Then

tempName = currDokName

'für Testläufe ausschalten

If File.Exists(InputOrdner & System.IO.Path.GetFileNameWithoutExtension(tempName) & ".ocr") Then

Kill(InputOrdner & System.IO.Path.GetFileNameWithoutExtension(tempName) & ".ocr")

End If

End If

'Originaldokument umbenennen in DokName

RenameDokFile(currDokName)

'Daten für ggf. weitere Zuordnung zwischenspeichern

FillDokInhalt()

With Start.currDoc

.Mandant = CInt(CbMandant.SelectedValue)

.Akte = CInt(CbAkte.SelectedValue)

.Vorgang = CInt(CbVorgang.SelectedValue)

'Prüfen, ob +1 erforderlich

.Dokument = currDokID

End With

'Wenn Attachment, in Tb Anlagen speichern

If isAttachment = True Then

Dim anyRow = \_FreeDMS\_StartDBDataSet.Anlagen.NewAnlagenRow

anyRow.DokID = lastID

anyRow.AnlagenID = (MaxID(CurrDB) + 1).ToString

\_FreeDMS\_StartDBDataSet.Anlagen.AddAnlagenRow(anyRow)

isAttachment = False

End If

'InhaltOCRTextBox.Text = tempOCRString

Try

Validate()

DokumenteBindingSource.EndEdit()

AnlagenBindingSource.EndEdit()

DokumenteTableAdapter.Update(\_FreeDMS\_StartDBDataSet.Dokumente)

AnlagenTableAdapter.Update(\_FreeDMS\_StartDBDataSet.Anlagen)

'erhöhen für dokNr.

Catch ex As Exception

MsgBox("Fehler beim Speichern!")

Exit Sub

End Try

currDokID = currDokID + 1

'Original-Doc umbenennen und im Archivordner speichern

tempOCRString = Nothing

RichTextBoxOCR.Clear()

TabControl1.SelectedTab = TabPage1

btWeiterZuordnung.Enabled = True

lbSpeichstatus.Text = "Datensatz gespeichert!"

lbSpeichstatus.ForeColor = Color.Green

speicherFlag = True

lboxSaveAtt.Enabled = True

LoadLvDok()

End Sub

Für die Speicherung in der Tab. Anlagen

If isAttachment = True Then

Dim anyRow = \_FreeDMS\_StartDBDataSet.Anlagen.NewAnlagenRow

anyRow.DokID = lastID

anyRow.AnlagenID = (MaxID(CurrDB) + 1).ToString

\_FreeDMS\_StartDBDataSet.Anlagen.AddAnlagenRow(anyRow)

isAttachment = False

End If

sind erforderlich:

1. Die ID des Mutterdokuments (i.d.R. der Mail) -> MailID
2. Die DocID der jeweils gespeicherten Anlage

Die MailID darf solange Anlagen gespeichert werden, nicht verändert werden.

MailID = CurrDokID – 1 geht also nicht!

Im Sub lvDoc\_Click die Variable „Mail“ mit CurrDokID füllen. Wird erst geändert, wenn ein neues Dokument bearbeitet wird, also nicht bei „Weiterer Zuordnung“ oder „SaveAtt“! -> Testlauf ok!

Fehler noch beim Aufruf docx in Anlage (Dokument ist noch geöffnet!)

Weitere Aufgaben:

* Übernommene Anlage in lboxSaveAtt Att löschen (Durchstrichen?)
* Anlagenliste in DocEdit einfügen
* Prüfen, ob bessere Bezeichnung von Mail und Anlage (Verbindung) möglich ist