

Handbuch „Fax/E-Mail Schnittstelle“

Ersteller: Cindy Kassab
Oliver Perschke
Dennis Röder

Stand 03.11.2023 Version 1.1

Änderungshistorie

Datum	Version	Verfasser	Änderungen
03.2023	0.1	Cindy Kassab	Initiales Setup
04.2023	0.2	Oliver Perschke	Anpassungen auf CKMail 2.0
06.2023	0.9	Oliver Perschke	Finalisierung und Übernahme der Konfigurationen aus den Umgebungen
06.2023	1.0	Alle	Finalisierung und Übergabe an AOK
03.11.23	1.1	Alle	KCFax Option Dokumentation hinzugefügt

Inhaltsverzeichnis

1	EINFÜHRUNG	7
1.1	Management Summary	7
1.2	Einleitung / Ausgangssituation / Beschreibung der Lösung	7
1.3	Abgrenzung	8
1.4	Das Konzept	8
1.5	Technische Voraussetzungen	8
2	GENERISCHES SETUP	9
2.1	Anlieferung:	9
2.2	Verarbeitung in NiFi:	9
2.3	Verarbeitung im Programm	9
2.3.1	CKMail:	9
2.3.2	KCFax:	10
2.4	Weiterleitung an Folgesysteme:	11
3	NIFI	11
3.1	NiFi:	11
3.2	Installation:	11
3.3	Konfigurationsschritte:	12
4	DER PROZESS IM SCHAUBILD:	13
5	DAS TOOL	14
5.1	Die Software Installation	15
5.2	Die Konfiguration	15
5.2.1	Die Datei App.Config	15
5.2.2	Die Datei log4net.config	18
5.3	Checklist für Changes	18
5.3.1	Versionsupdate	18
5.3.2	Datenupdate (Neue Daten einspielen)	18
5.3.3	Umgebungsanpassung	18
5.3.4	Anpassung von Konvertierungsparametern	19
5.3.5	Datenbankupdate RDBMS (SQL/Oracle Sever)	19
5.4	Die Datenbanken	19
5.4.1	Die SQL Server Datenbank	19
5.4.2	Die Oracle Reporting Datenbank	22
6	DIE TESTUMGEBUNG	23
6.1	Der physikalische Server	23
6.2	Der NiFi Server	24
6.3	Das Programm	24
6.3.1	CKMail	24
6.3.2	KCFax	25
6.3.3	Die Konfiguration des KCFax	25
6.4	Die Weitergabe	25
7	DIE PRODUKTIONSUMGEBUNG	25
7.1	Der physikalische Server	25
7.2	Der NiFi Server	26
7.3	Das Programm	26
7.3.1	CKMail	26

7.3.2	KCFax	26
7.4	Die Weitergabe	27
8	DER CHANGE PROZESS	27
8.1	Der Anpassungsprozess für Änderungen der Datenbanktabellen	27
8.1.1	Anlegung einer neuen Dokuklasse	27
8.1.2	Löschen einer Dokuklasse	27
8.1.3	Änderung formeller Prüfungen bestehender Dokuklassen	27
8.1.4	Technische Aktivitäten:	28
9	REFERENZEN	28
9.1	Der Anpassungsprozess für Änderungen der Datenbanktabellen	28
9.2	Datenbanklogin Prod:	28
9.3	Erstellung eines verschlüsselten Passwords:	28
9.4	Postfächer:	29
9.5	Das Programm CKMail.exe im Detail	30
9.5.1	Klasse Programm (ProgramMain.cs und NonMain.cs)	33
9.5.2	Die Klassen zur Verarbeitung der E-Mail:	35
9.5.3	Die Klasse CheckRun:	35
9.5.4	Die Klassen in der Bibliothek DataAccess:	35
9.5.5	Die Konverter:	36
	PDF Dokumente:	36
	PNG Images:	36
	JPG Images:	36
	TIF/TIFF Images:	36
	Text Dokumente:	36
9.5.6	Die Bibliotheken:	36
9.5.7	Die KonfigDatei App.Config	37
9.5.8	Die Klassenreferenzen	37
9.5.9	Das Programm ImageMagick	37
9.5.10	Das Programm Repository CKMail	37
9.5.11	Das Programm Repository KCFax	37
9.6	Rückweisungsgründe	37
9.6.1	Referenz:	38
9.6.2	Erklärung:	38
9.7	Die Unterschiede von KCFax zum Programm KCFax.exe im Detail	40

Abbildungsverzeichnis

Abbildung 1 Das Prozess Schaubild	8
Abbildung 2: NIFI Flow Schaubild (Beispiel)	12
Abbildung 3: Prozessschaubild CKMail.....	14
Abbildung 4: Datenbank NiFi Sendungsnummern	19
Abbildung 5: Datenbank AttachmentRules.....	20
Abbildung 6: Datenbank Regelwerk.....	20
Abbildung 7: Datenbank Rückweisungsgründe.....	21
Abbildung 8: Datenbank Übersetzungstabelle.....	21
Abbildung 9: Der Reporting Stapel.....	22
Abbildung 10: Die Reporting Sendung	22
Abbildung 11: Das Reporting Dokument.....	23
Abbildung 12: Die Tabelle Abweisung.....	23
Abbildung 13: Der Change Prozess.....	27
Abbildung 14: Outlook Login	30
Abbildung 15: Bibliotheken in CKMail	31
Abbildung 16: Klassenstruktur CKMail	32
Abbildung 17: Klassenstruktur DataAccess	32
Abbildung 18: Klassenstruktur Hilfsbibliotheken	33

Tabellenverzeichnis

Tabelle 1: Die E-Mail-Postfächer der Umgebungen	7
Tabelle 2: Server in der Testumgebung	24
Tabelle 3: Nifi Server in der Testumgebung	24
Tabelle 4: CKMail in der Testumgebung	24
Tabelle 5: Ausgabeschnittstelle CKMail in der Testumgebung	25
Tabelle 6: Server in der Produktionsumgebung	26
Tabelle 7: Nifi Server in der Produktionsumgebung	26
Tabelle 8: CKMail in der Produktionsumgebung	26
Tabelle 9: : Ausgabeschnittstelle CKMail in der Produktionsumgebung	27
Tabelle 10: eePV Dokuklassen	28
Tabelle 11: APD Dokuklassen	28
Tabelle 12: Postfächer	Fehler! Textmarke nicht definiert.

1 Einführung

Dieses Dokument fasst alle Ergebnisse und Teilergebnisse der Analyse- und Implementierungsphase bei der AOK Sachsen-Anhalt zusammen und erläutert die Ergebnisse und beschreibt die durchgeführten Maßnahmen.

Außerdem werden in diesem Dokument alle analysierten Prozesse noch einmal beschrieben, bzw. die bereits vorhandenen Beschreibungen zusammengefasst, um diese in einem Gesamtdokument zu bündeln. So dient dieses Dokument nicht nur für die weitere Optimierung des Fax/E-Mail Schnittstelle Verarbeitung im RZ Magdeburg, sondern kann in Auszügen auch als Leitfaden für neue Mitarbeiter verwendet werden.

Im Rahmen der Einführung eines neuen Input-Managementsystems und der Einführung neuer Logikmodule kann das Dokument zudem als Dokumentation des Ist-Standes angesehen werden und als Grundlage für die Fax-/E-Mail und Kundencenter Fax (KCFax) Schnittstelle dienen.

1.1 Management Summary

In der AOK Sachsen-Anhalt existiert ein Inputmanagement-Prozess zur Verarbeitung eingehender Post und Faxe. Im Kontext dieser Prozesse wird eine Fax-/E-Mail-Schnittstelle eingesetzt, welche den nachverarbeitenden Systemen die per E-Mail oder Fax eingehende Post zuführt.

Dabei wird Es wird ein Maximalvolumen von 2500 E-Mails / Arbeitstag erwartet. Dieses wird innerhalb der Servicezeiten 06:00Uhr – 20:00Uhr auftreten und muss in der Folge tagesaktuell verarbeitet werden. Die Verarbeitung (Zuführung der E-Mails) erfolgt dabei automatisiert und ohne menschliche Interaktion.

1.2 Einleitung / Ausgangssituation / Beschreibung der Lösung

Die AOK Sachsen-Anhalt (bzw. deren Dienstleister) hat zurzeit ein neues IPM-System erstellt und betreibt dazu verschiedene Datenverarbeitungs- und Transferlösungen.

Dieses Dokument beschreibt die für diesen Zweck zu implementierende Fax-/E-Mail-Schnittstelle als Interface zwischen Dokumenteneingang und dem CrossCap System zur Validierung der Eingänge.

Ausgangslage:

Verschiedene Mitarbeiter der AOK versenden zwecks Archivierung Dokumente als E-Mail-Anhang an zwei definierte E-Mail-Postfächer. Die eingehenden E-Mails sollen einer fachlichen und formalen Prüfung unterzogen werden. Dazu dient die erstellte Fax-/E-Mail-Schnittstelle.

Dazu werden E-Mails aus den für jede Umgebung definierten Postfächern abgeholt, formal und technisch geprüft und der manuellen Validierung der jeweiligen Umgebungen zugeführt.

Umgebung	Adresse
Produktion	prod.epost@san.aok.de
Test	qa.epost@aoksanint.de
Produktion	prod.latescan@san.aok.de
Test	qa.latescan@aoksanint.de
Produktion	ipm.no-reply@san.aok.de
Test	ipm.no-reply@aoksanint.de
Produktion KCFax	
Test KCFax	

Tabelle 1: Die E-Mail-Postfächer der Umgebungen

Die Bearbeitung der E-Mails erfolgt im Regelfall komplett ohne menschliche Interaktion (Dunkelverarbeitung). Nur im Fehlerfall ist ein manueller Eingriff vorgesehen.

1.3 Abgrenzung

Dieses Dokument beschreibt ausschließlich die Verarbeitung der Email in der Dunkelverarbeitung. Die Anlieferung der Emails durch Sachbearbeiter und Bereitstellung der Emails im Filesystem des Verarbeitungsrechners sowie die Weiterleitung an die Folgesysteme sind nicht Bestandteil dieses Dokumentes und sind im DV-Gesamtkonzept beschrieben.

1.4 Das Konzept

Die Entwicklung und Implementierung folgt der in der AOK üblichen Methodik des Workflowmanagementtools NiFi. Das bedeutet, die Verarbeitung der angelieferten EML Dateien erfolgt durch Abholung von NiFi, Weiterleitung an das Verarbeitungstool und Ausgabe in ein definiertes Ausgabeverzeichnis aus dem diese dann weitergeleitet werden. Dies erfolgt durch ein Kopierprogramm und damit Weiterleitung an das Folgesystem (CrossCap).

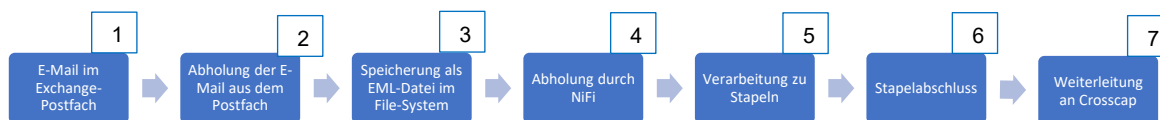


Abbildung 1 Das Prozess Schaubild

Die Punkte 4,5 und 6 sind Bestandteil dieser Dokumentation.

1.5 Technische Voraussetzungen

Für die Installation der Anwendung sind folgende Voraussetzung zwingend erforderlich:

- Ein Server mit Windows Server 2016 und neuer
- Installation einer Java Runtime Umgebung Version 8 oder 11 (alternativ eine aktuelle Version OpenJAVA)
- .Net Runtime Umgebung .NET 5 und neuer
- Eine Datenbankinstanz in einem SQL-Server zur lokalen Datenhaltung
- Eine Datenbankinstanz in einem Oracle Server für das Reporting
- Eine Installation von Imagemagick
- Ein Mail-Server der als Relay für den Empfang und die Weiterleitung von Emails agieren kann
 - Dieser muss:
 - Über das SMTP-Protokoll ansprechbar sein (Zur Weiterleitung von Admin und Rücksendemails)
 - Berechtigungsmaßig zur Weiterleitung konfiguriert sein

Sinnvolle zusätzliche Tools sind:

- Notepad++ zur Editierung und Analyse von Daten und Protokollen
- Outlook zur Anzeige von EML-Dateien
- Ein Monitoring zur Überwachung der Ergebnisse und Fehlerzustände
- Ein Tool/Skript zum Kopieren der Dateien in die Folgeschnittstellen

Extern benötigt:

- Visual Studio 2019 oder neuer zur Analyse und Erweiterung/Anpassung des Sourcecodes

Das Verarbeitungsprogramm hat folgende Voraussetzungen:

- Das Programm ist selbstständig lauffähig unter allen .NET unterstützenden Betriebssystemen
- Es wird gesteuert durch ein beliebiges Prozessmanagementtool (Hier: NiFi in aktueller Version zurzeit 1.19.1)

2 Generisches Setup

2.1 Anlieferung:

Die Schnittstelle, die zur Anlieferung der EML-Dateien verwendet wird, ist der IMAP-Kollektor der Firma SCE. Dieser sammelt die Emails aus dem Exchange Server der AOK ein und stellt diese in einem beliebigen Verzeichnis zur Verfügung. Des Weiteren sind die täglichen Laufzeiten frei konfigurierbar. Zusammenfassung:

Die Anlieferung stellt EML-Dateien in einem Verzeichnis zur Verarbeitung bereit.

2.2 Verarbeitung in NiFi:

Die installierte NiFi Instanz wird benötigt, um die bereitgestellten EML-Dateien der Verarbeitung durch das CKMail zur Verfügung zu stellen, das CKMail zu starten und ihm dabei den Namen des zu verarbeitenden Files zu übergeben sowie das Ergebnis zu überwachen und gegebenenfalls einen Error auszulösen.

Zusammenfassung:

NiFi sammelt die angelieferten EML-Dateien ein, übergibt sie an das CKMail und initiiert ggfls. Fehlermeldungen aus.

2.3 Verarbeitung im Programm

2.3.1 CKMail:

Das Programm CKMail wurde in C# mit dem .NET Core Framework entwickelt und hat folgende Aufgaben.

- Verarbeitung einer übergebenen EML-Datei
 - o Analyse der Betreffzeile
 - o Für bestimmter Emails wird eine Verzögerung der Verarbeitung um 24h ausgelöst
 - o Fachliche Überprüfungen
 - o Konvertierung von Anhängen in andere Dateiformate
 - o Rückweisung bei fachlich fehlerhaften Emails
 - o Ablage korrekter Dateien in einem Stapelsystem
 - o Technisches Fehlerhandling
- Abschluss des Stapelsystems
 - o Beim Erreichen eines konfigurierbaren Alters des Stapels
 - o Beim Erreichen einer konfigurierbaren Größe des Stapels
 - o Ablage der Stapelstruktur in der Reporting Datenbank

Die Installation erfolgt in einem beliebigen Verzeichnis.

Abseits dieses Verzeichnisses müssen einige weitere Verzeichnisse eingerichtet werden, damit die Verarbeitung erfolgreich laufen kann. Die Existenz und Zugreifbarkeit dieser Verzeichnisse wird bei jedem Programmstart überprüft.

Dabei handelt es sich um Verzeichnisse, die in der Konfigurationsdatei beliebig gewählt werden können.

Des Weiteren müssen folgende Elemente in der Konfigurationsdatei eingetragen werden:

- die Datenbankverbindungen zum NiFi SQL Server
- die Datenbankverbindungen Oracle Reporting Server
- die Konfiguration des versendenden Emailservers (Relay host)
- die maximale Größe des Stapels in Anzahl Sendungen
- das maximale Alter des Stapels in Minuten
- die Rückhaltezeit der Zugewinnmails
- die Konvertierungsoptionen für Imagemagick
- die Umgebungskennung für die Stapelbenennung

Zusammenfassung:

Das CKMail verarbeitet angelieferte EML-Dateien, legt die Ergebnisse in einer Stapelstruktur ab und bereitet sie für die Weiterleitung vor, d.h. die Stapel erhalten eine eindeutige Kennung, die dem Folgesystem mitteilt, das der Stapel zur Abholung bereit ist.

2.3.2 KCFax:

Das Programm KCFax wurde in C# mit dem .NET Core Framework entwickelt und hat folgende Aufgaben.

- Verarbeitung einer übergebenen EML-Datei
 - o Analyse der Betreffzeile
 - o Fachliche Überprüfungen
 - o Konvertierung von Anhängen in andere Dateiformate
 - o Abweisung mit einer Fehlermail an den Admin bei fehlerhaften Dateien
 - o Ablage korrekter Dateien in einem Stapelsystem
 - o Technisches Fehlerhandling
- Abschluss des Stapelsystems
 - o Beim Erreichen eines konfigurierbaren Alters des Stapels
 - o Beim Erreichen einer konfigurierbaren Größe des Stapels
 - o Ablage der Stapelstruktur in der Reporting Datenbank

Die Installation erfolgt in einem beliebigen Verzeichnis.

Abseits dieses Verzeichnisses müssen einige weitere Verzeichnisse eingerichtet werden, damit die Verarbeitung erfolgreich laufen kann. Die Existenz und Zugreifbarkeit dieser Verzeichnisse wird bei jedem Programmstart überprüft.

Dabei handelt es sich um Verzeichnisse, die in der Konfigurationsdatei beliebig gewählt werden können.

Des Weiteren müssen folgende Elemente in der Konfigurationsdatei eingetragen werden:

- die Datenbankverbindungen zum NiFi SQL Server
- die Datenbankverbindungen Oracle Reporting Server
- die Konfiguration des versendenden Emailservers (Relay host)
- die maximale Größe des Stapels in Anzahl Sendungen
- das maximale Alter des Stapels in Minuten
- die Rückhaltezeit der Zugewinnmails
- die Konvertierungsoptionen für Imagemagick
- die Umgebungskennung für die Stapelbenennung

Zusammenfassung:

Das CKMail/KCFax verarbeitet angelieferte EML-Dateien, legt die Ergebnisse in einer Stapelstruktur ab und bereitet sie für die Weiterleitung vor, d.h. die Stapel erhalten eine eindeutige Kennung, die dem Folgesystem mitteilt, das der Stapel zur Abholung bereit ist.

2.4 Weiterleitung an Folgesysteme:

Zur Weiterleitung aus dem Stapelverzeichnis wird zurzeit das Tool MonsterCopy verwendet. Dieses wird durch die Windowsaufgabensteuerung angestoßen und kopiert die abgeschlossenen Verzeichnisse in den CrossCap Eingangsordner, der auf dem Windows Server zu diesem Zweck gemountet sein muss.

Zusammenfassung: Es wird ein Kopiertool verwendet um die abgeschlossenen Stapel an das nächste Tool weiterzuleiten.

3 NiFi

3.1 NiFi:

Apache NiFi ist als Big Data System ein Tool für reibungslosen Datenfluss. Dataflows sind Prozessketten, die Daten entgegennehmen, verarbeiten und weiterleiten. Ein einfaches Beispiel ist die automatische Einordnung von E-Mails. Weitere Beispiele wären Geschäftsabläufe zu Optimieren oder das Kundenverhalten auszuwerten, um Kosten zu senken, Risiken zu minimieren und die Gewinne zu vergrößern.

Apache NiFi ist ein Tool, das Daten nicht nur aus unterschiedlichen Quellen einsammeln kann, sondern im Standardumfang zahlreiche Transformationsmöglichkeiten bietet und viele unterschiedliche Ausgabekanäle. Bei diesem Projekt geht es sehr oft um die Einsammlung, Verarbeitung und Verteilung von XML-Daten.

Des Weiteren bietet Apache NiFi ein umfangreiches Angebot an Prozessoren für sehr viele Schnittstellen und Transformationen. NiFi bietet dabei ein Graphisches Frontend mit den zugehörigen Prozessoren. Es ist komplett Open Source und in Java geschrieben. Somit ist auch eine manuelle Erweiterung oder Anpassung jederzeit möglich.

Zum Betrieb der NiFi Anwendung ist eine „Java Runtime Umgebung“ notwendig. Gegenwärtig werden die Version 8 und Version 11 unterstützt. Will man allerdings auch das Registry einsetzen, so empfiehlt sich Java 8 JRE in der aktuellen Fassung.

3.2 Installation:

Die Installation der NiFi Umgebung setzt mindestens, wie bereits beschrieben, eine JRE der Version 8 voraus. Das Setzen der Variablen JAVA_HOME ist sehr sinnvoll. Alternativ kann die Datei java.exe auch in den PATH mit aufgenommen werden.

Ist Java erfolgreich eingerichtet, so kann das Installationsfile für NiFi in einem Verzeichnis abgelegt und ausgepackt werden (.zip Files).

Sind keine weiteren Sicherheitsoptionen gewünscht, so kann NiFi direkt gestartet werden.

Dies geschieht mittels eines Skripts:

Dazu wird das ZIP-Archiv von NiFi in einem beliebigen Verzeichnis ausgepackt und die Datei:

```
# cd Nifi folder\bin
# run-nifi.bat
```

ausgeführt. Beim ersten Starten konfiguriert dieses Programm dabei alle notwendigen weiteren Arbeiten und generiert einen Benutzer und ein Passwort und startet den NiFi Server in einer DOS Box (cmd). Mit den generierten Daten kann jetzt in einem beliebigen Browser über die URL: <https://localhost:8443/nifi> auf die grafische Benutzeroberfläche zugegriffen werden.

3.3 Konfigurationsschritte:

CKMail und KCFax wiesen identische Prozessschritte auf und unterscheiden sich nur in den absoluten Pfaden. Die Installation der Binaries ist identisch, sie sind nur durch Pfade getrennt. Beider Programme laufen aber unabhängig voneinander.

Nach dem ersten Start der Applikation ist ein User mit einem Password erstellt worden. Die zugehörigen Daten sind dem NiFi-Logfile (unter NiFiHome/logs/nifi-app.log) zu entnehmen.

Warnung: Werden diese Daten nicht manuell gesichert, rollieren Sie mit der Zeit dem Log heraus und gehen damit verloren.

Diese Daten müssen genutzt werden, um alle weiteren Arbeiten auf der GUI durchzuführen.

Damit die folgenden Schritte funktionieren, muss das CKMail bereits wie unter 3.2 beschrieben, installiert sein.

Anzulegen ist nun eine Prozessgruppe mit dem Namen Fax-/E-Mail Schnittstelle:

Darin sind anzulegen:

- Ein „GetFile“ Prozessor um jede Minute eine Datei mit dem Namen „check“ aus dem Verzeichnis basecheck in das Input Verzeichnis zu kopieren, um damit die Routine der Filesystemkontrolle zu starten. Dabei bleibt die Datei im „check“-Verzeichnis erhalten. Der Inputfolder wird in diesem NiFi Folder definiert und lautet zurzeit C:\NiFiFaxEMail\base\input
Dies triggert die Überprüfung des Abschlusses der Ablagefolder mit der Endung „rdy“ zur Weiterverarbeitung mit einen beliebigen Kopierprogramm nach entweder x E-Mail oder y min Alter der Verzeichnisse.
- Ein GetFile Prozessor für das Abholen der E-Mails aus den vorgegebenen Übergabeverzeichnissen des IMAP-Kollektors.
- Ein GetFile Prozessor der mittels E-Mail oder Check File Verarbeitungsprozess anstößt.
- Ein ExecuteStreamProzessor der das eigentliche Programm startet und die Umgebung verwaltet
- Ein PutFile Prozessor zur Ablage der Kopie der originalen E-Mail (Sicherheitskopie im Originalfolder)
- Ein PutFile Prozessor zur Ablage eines ggfs. Auftretenden technischen Fehler (Errorhandling (Ablage Datei im Errorfolder))

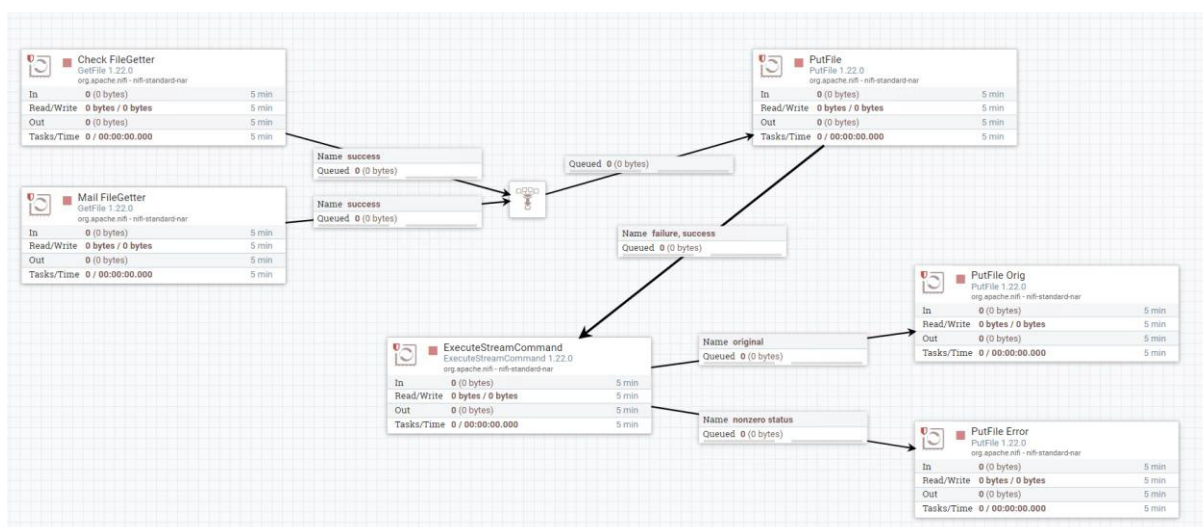


Abbildung 2: NIFI Flow Schaubild (Beispiel)

Nach der Aktivierung dieser Prozessgruppe kann nun automatisiert die Verarbeitung durchgeführt werden.

```
# cd Nififolder\bin
# run-nifi.bat
```

Weitere Konfigurationsmöglichkeiten sind nicht notwendig. Es ist aber möglich weitere User und Einstellungen durchzuführen. Als Referenz ist dazu die NiFi Dokumentation unter (s.u.) zu verwenden.

Apache NiFi Walkthroughs Allgemeine Informationen:
<https://nifi.apache.org/docs/nifi-docs/html/walkthroughs.html>

Security Konfiguration:
<https://nifi.apache.org/docs/nifi-docs/html/walkthroughs.html#securing-nifi-with-tls-toolkit>

Der allgemeine Admin Guide
<https://nifi.apache.org/docs/nifi-docs/html/administration-guide.html>

User Identifikation:
https://nifi.apache.org/docs/nifi-docs/html/administration-guide.html#user_authentication

Alles andere zu NiFi:
<https://nifi.apache.org/docs.html>

4 Der Prozess im Schaubild:

Die Anlieferung:

Der IMAP Kollektor liefert alle Emails als .eml Files im Eingangsverzeichnis ein (wird definiert im GetFile Prozessor von NiFi und im Ausgabeordner des IMAP Kollektors).

Die Abholung:

Die Abholung der Emails aus dem Eingangsordner wird durch einen GetFile Prozessor in NiFi realisiert. Ein zweiter GetFile Prozessor steuert jede Minute eine Datei mit dem Namen „check“ ein.

Über einen Funnel werden beide Dateiarnten an den InputProzessor weitergeleitet, der mittels eines ExecuteStream Prozessors das verarbeitende Programm aufruft und ihm einen Dateinamen mitgibt. Dieser Dateinamen repräsentiert das zur verarbeitende „eml“ oder „check“-File.

Der Aufruf des Mailverarbeitungsprogramms

Der ExecuteStream Prozessor führt das Programm für jede übergebene Datei exakt einmal aus und übergibt das Ergebnis je nach Erfolg der Verarbeitung entweder an den PutFile Prozessor für ERROR Files (Rückgabewert des Programms != 0) oder den PutFileProzessor für das Originalfile (Rückgabewert des Programms =0).

Während der Verarbeitung werden verschieden Verzeichnisse, die vorhanden sein müssen, benutzt.

OutputFolder => Der Ordner, in dem die Stapel gebildet werden
EmailOutFolder => Der Ordner in dem die versandten Abweisungs-mails liegen
BackupFolder => Das Backup, das beim Stapelabschluss gebildet wird
WorkFolder => Das Arbeitsverzeichnis, in dem die Konvertierung stattfindet. Dieses wird bei jedem Programmaufruf gelöscht um eine saubere Verarbeitung zu gewährleisten.

Die Konfiguration des Programms liegt in der App.config und die Verzeichnisse sind abhängig von der Umgebung und müssen für jede Umgebung einzeln angepasst werden.

Die Ausgabe in Form eines Stapels:

Nach erfolgreicher Verarbeitung der einzelnen Mails liegen diese als Sendung in Form eines Stapels vor. Bei jedem Programmaufruf wird getestet, ob der Stapel abgeschlossen werden muss. Ist dies der Fall, erfolgt ein Stapelabschluss vor der Verarbeitung der Email. D.h. die folgende Email ist dann die erste Sendung in einem neuen Stapel.

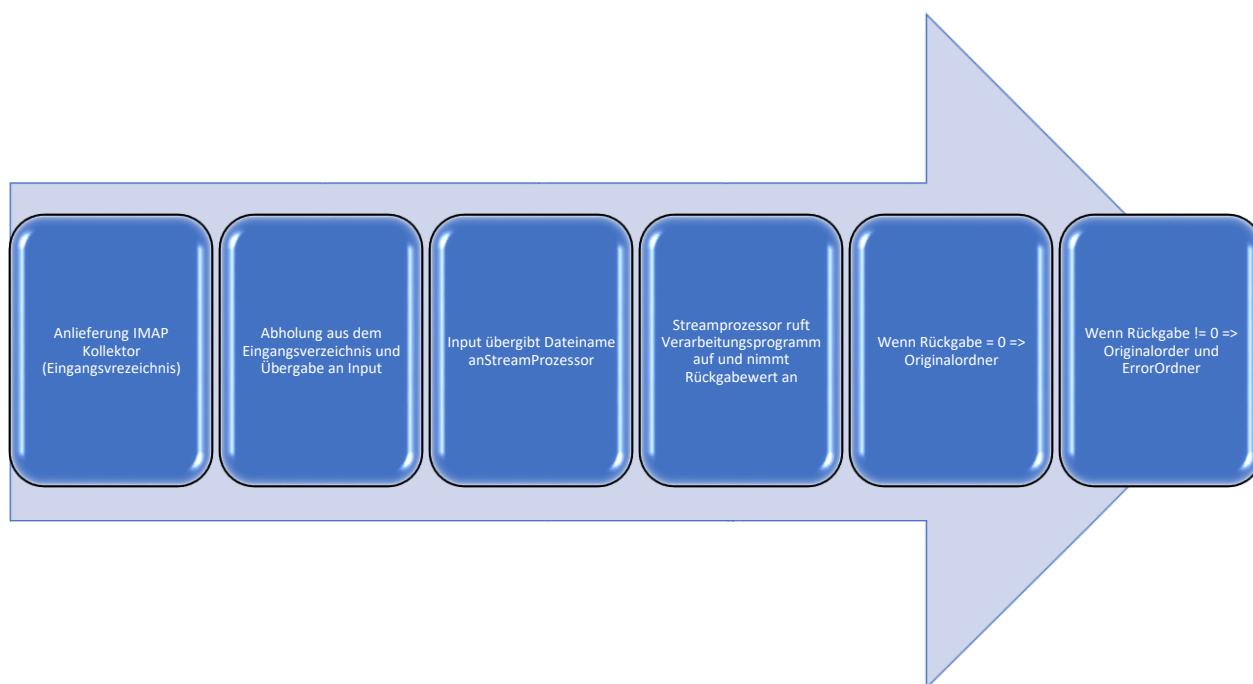


Abbildung 3: Prozessschabild CKMail

5 Das Tool

Allgemein: Das Programm nutzt das .Net Core Framework 5.0 und das Entity Framework Core 5 zur Datenverarbeitung.

Für Loggingzwecke wird das Log4Net Framework angewandt.

Gegebenenfalls erzeugte E-Mails werden durch das MimeKit und das MailKit erzeugt, bzw. versandt. Die HTML-Unterstützung erfolgt durch das HtmlAgilityPack.

Die Programme Programm CKMail/KCFax haben zwei Verarbeitungsmodi:

1. Verarbeitung einer E-Mail, deren Dateiname dem Programm übergeben wird. Die Ergebnisse werden im Ausgabeverzeichnis zu Stapeln zusammengefasst.
Dieser Modus löst außerdem auch den Stapelabschlussmodus aus, der vor der Mailverarbeitung abgeschlossen wird. Dies stellt sicher, dass ein Stapel nie mehr als 30 Sendungen enthalten kann, da vor der Mailverarbeitung die Stapelgröße überprüft wird.
2. Stapelabschlusscheckmodus: Um diesen Modus zu starten wird dem Programm eine Datei namens „check“ mitgegeben. Diese kann in einem beliebigen Verzeichnis liegen (Die Übergabe muss den vollen Dateinamen enthalten). Der Inhalt der Datei spielt keine Rolle. Sie kann auch 0 Byte groß sein. Dieser Modus prüft die Größe und das Alter der Stapelverzeichnisse und bereitet die Verzeichnisse ggfls. auf die Weiterleitung von, indem er die Endung „.rdy“ and den Stapelnamen hängt.

Der Checkrun ist notwendig, da es nicht sichergestellt ist, dass in einen festgelegten Zeitraum wenigstens eine E-Mail verarbeitet wird. Da das Stapelalter aber unabhängig von der Größe auch ein Maximalalter hat, muss dieses unabhängig von der Mailverarbeitung überprüft und gegebenenfalls auch verarbeitet werden.

5.1 Die Software Installation

Die Installation der CKMail/KCFax Applikation erfolgt durch einfache Ablage der Binaries in einem beliebigen Verzeichnis. Im nächsten Schritt sind die Elemente in der Dateien App.config und Log4net.config anzupassen.

Danach kann ein manueller Testlauf erfolgen, entweder mit einer E-Mail oder mit der Checkdatei. Danach sollte das Logfile auf Funktion überprüft werden.

Als letztes sollte dann einmal der automatisierte Ablauf mit NiFi getestet werden.

5.2 Die Konfiguration

5.2.1 Die Datei App.Config

Die Konfiguration des Programms ist zentral in der Konfigurationsdatei App.config im Installationsverzeichnis abgelegt (Identisch für beide Varianten)
In dieser Datei werden alle umgebungsspezifischen Parameter hinterlegt und diese muss für jede Instanz individuell angepasst werden.

Die Entwicklung und Implementierung folgt der in der AOK üblichen Methodik des Workflowmanagementtools NiFi. Es wird eine NiFi Umgebung mit den folgenden Ausgabebereichen eingerichtet:



Erklärung der Schlüssel (identisch für beide Varianten, KXFax ist nur einfacher):

Der Block Verzeichnisse:

```
<add key="OutputFolder" value="U:\WorkArea\CKMail\VerzeichnisMitStapeln"/>
<add key="EmailOutFolder" value="U:\WorkArea\CKMail\MailOut"/>
<add key="BackupFolder" value="U:\WorkArea\CKMail\Backup"/>
<add key="Workfolder" value="U:\WorkArea\CKMail\Workfolder"/>
<add key="ConverterExePath" value="U:\WorkArea\NiFiCMTTestumg\imagemagick\convert.exe"/>
<add key="GhostScriptPath" value="U:\WorkArea\NiFiCMTTestumg\imagemagick"/>
```

OutputFolder:

Dies ist der Ausgabefolder aus dem MonsterCopy die Ausgabe abholt.

EmailOutFolder:

In diesem Folder werden die Abweisungs-mails generiert, aus diesem versandt und verbleiben dort bis zur manuellen Löschung.

BackupFolder:

Abzuschließende Stapel werden in diesem Ordner vor der Verarbeitung gesichert. Diese kann (und sollte wegen des Platzbedarfs) nach Belieben aufgeräumt werden. Dies kann und sollte automatisiert erfolgen.

Workfolder:

Dies ist der Arbeitsordner in dem die Datenkonvertierung durchgeführt wird. Nach erfolgreicher Konvertierung werden die Daten in das Ausgabeverzeichnis () kopiert. Dieses Verzeichnis wird am Programmende gelöscht und somit verbleiben keine Artefakte im Verzeichnis. Im Fehlerfall werden die Daten nicht kopiert und eine Mail an den Administrator => „AdminMail“ versandt.

ConverterExePath:

Die ist der komplette Pfad zum Konverter der Imagemagick Suite. Dieses Programm wird zur Konvertierung der Datenformate benötigt.

GhostScriptPath:

Für die Ermittlung der Seitenzahl von PDF Dokumenten wird die GhostScript DLL benötigt. Sie ist Bestandteil von Imagemagick. Damit diese auch unabhängig von Imagemagick verwendet werden kann, gibt es diesen Eintrag.

Die Stapelverarbeitungseinstellungen:

Maximales Alter der Stapel in Minuten:

```
<add key="StapelZeit" value="30"/>
```

Maximale Größe der Stapel in Sendungen:

```
<add key="StapelSize" value="30"/>
```

Die Datenbankverbindungen:

Der SQL Server Datenbankuser. Wird vom DBA festgelegt.

```
<add key="SqlUser" value="SVC_NiFi"/>
```

Das Password in verschlüsselter Form

```
<add key="SqlPasswort" value="gecrpyted Password"/>
```

Der SQL Server Connectionstring

```
<add key="connectionstring" value="sqlserver_connstring"/>
```

Der SQL Server Connectionstring

```
<add key="OraReport_Connector" value="oracle connstring" />
```

Die Mailrelay-Parameter werden benötigt, um Abweisungs-mails oder technische Informationen zu versenden:

User und Passwprd können bei Bedarf optional zugefügt werden.

```
<add key="mailServer" value="MailServer.AOKSAN.AOK"/>
```

```
<add key="mailPort" value="25"/>
```

```
<add key="mailUser" value=""/>
```

```
<add key="mailPassword" value=""/>
```

```
<add key="AdminMail" value="empfaengerEmail"/>
```

Die Parameter für die Verzögerung spezifischer Emails (nur CKMail) um einen bestimmten Zeitraum

Zu verzögernde Dokuklassen in kommaseparierter Liste

```
<add key="Delay24hFor" value="YLAMIXXXXXX, YLAMIYYYYY"/>
```

Der Pfad zur Zwischenspeicherung

```
<add key="Delay24hFolder" value="Pfad\Delay24h"/>
```

Der Pfad zur erneuten Einspielung der Emails (für gewöhnlich der Ablagepfad des IMPA Kollektors)

<add key="ReturnFolder" value="Pfad\MailIn"/>

Anzahl der Stunden, die die Verzögerung für oben genannte Dokuklassen dauern soll.

<add key="HoursForDelay" value="24"/>

Allgemeine Einstellungen:

Der Pfad zur Datei convert.exe zur externen Dateikonvertierung

<add key="convertPath" value="Pfad\imagemagick\convert.exe"/>

Sollen Sonderzeichen in der Betreffzeile eliminiert werden?

<add key="SonderZeichenEscape" value="Nein"/>

Das Umgebungskennungselement in den Stapelnamen

<add key="Umgebungskennung" value="A1"/>

Die Zeile From: in der Abweisungsmail.

<add key="fromMail" value="no-reply@aoksanint.de"/>

Verzeichnisse mit diesen Kennungen werden vom Programm nicht abgeschlossen.

(Zur Vermeidung von Interaktionen mit anderen Prozessen)

<add key="ClosingFolderNameExceptions" value="extensions"/>

Die Parameter die Imagemagick verwendet, um den genannten Dokumententyp zu konvertieren.

<add key="PDFConverterOpts" value="PDF Conf Parameter"/>

- Nur CKMail

<add key="JPGConverterOpts" value="JPG Conf Parameter "/>

<add key="TextConverterOpts" value="Text Conf Parameter "/>

<add key="TIFFConverterOpts" value="TIFF Conf Parameter "/>

Soll eine Konvertierung für einen bestimmten Parameter durchgeführt werden (An/Ausschalter Ja/Nein)

<add key="PDFConvOn" value="Ja"/>

- Nur CKMail

<add key="JPGConvOn" value="Ja"/>

<add key="TextConvOn" value="Ja"/>

<add key="TIFFConvOn" value="Ja"/>

Andere Parameter

Die Kennung an der der abgeschlossen Stapelordner erkennbar ist

<add key="Finalkennung_Folder" value=".rdy"/>

Soll das Reporting in der Datenbank benutzt werden (Ja/Nein)

<add key="ReportingAktiv" value="Nein"/>

Eine neue Paramteroption für die Konvertierung des Mailbody in ein Tiff Dokument

<add key="TextToPDFOptionsForRejectMail" value="-page a4 -density 300"/>

Maximale Anzahl von Seiten in einem PDF Dokument. Eine Seitenzahl von > X führt zu einer Abweisungsmail

<add key="PDFMaxPageCount" value="250"/>

Eine Option zum Abschalten der Abweisungsmail. Für automatisierte Tests

```
<add key="MailVersand" value="Ja"/>
```

Soll vor dem Stapelabschluss ein Backup des Stapels durchgeführt werden? (Ja/Nein)

```
<add key="BackupAktiv" value="Nein"/>
```

Die Datei App.config wird bei jedem Programmstart erneut eingelesen und interpretiert. Somit sind Anpassungen zu jeder Zeit möglich und sofort aktiv.

5.2.2 Die Datei log4net.config

Die Konfigurationsdatei für das Logging ist die Datei log4net.config. (Identisch für beide Varianten) Diese liegt im Root Verzeichnis der Anwendung und enthält nur Informationen die für das Logging relevant sind



log4net.config

Die Referenz alle dieser Parameter ist zu finden unter:

[Log4net Tutorial for .NET Logging: 14 Best Practices and Examples \(stackify.com\)](http://log4net.sourceforge.net/tutorial/14BestPracticesAndExamples.html)

Und

[Apache log4net – Apache log4net Manual: Configuration - Apache log4net](http://log4net.sourceforge.net/manual/configuration.html)

Für diese Applikation ist primär der Eintrag:

```
<file value="U:\WorkArea\CKMail\Protokoll\protokoll.log" />
```

relevant. Der hier aufgeführte Pfad zeigt auf den Ablagepunkt der Protokolldatei und kann beliebig angepasst werden, solange das Verzeichnis, das er enthält, vorhanden ist. Die anderen Parameter können bei Bedarf angepasst werden.

5.3 Checklist für Changes

5.3.1 Versionsupdate

Ein Versionsupdate ist ein kompletter Austausch der Binaries der Applikation. Dabei wird die Konfigurationsdatei der alten Version übernommen.

Gegebenenfalls sind dabei Anpassungen in der App.config durchzuführen. Dies muss bei gestoppter Verarbeitung erfolgen.

5.3.2 Datenupdate (Neue Daten einspielen)

Ein Datenupdate kann jederzeit eingespielt werden, da die Datenbank mit jedem Programmstart eingelesen wird. Es gibt keine Verbindung zwischen den verschiedenen Programmläufen. Es gibt aber mögliche Abhängigkeiten zu der Stapelnummer und Sendungsnummer. Diese sollten nicht manuell angepasst werden.

5.3.3 Umgebungsanpassung

Die Datei App.config wird bei jedem Programmstart erneut eingelesen und interpretiert. Somit sind Anpassungen zu jeder Zeit möglich und sofort aktiv.

5.3.4 Anpassung von Konvertierungsparametern

Die Datei App.config wird bei jedem Programmstart erneut eingelesen und interpretiert. Somit sind Anpassungen zu jeder Zeit möglich und sofort aktiv.

5.3.5 Datenbankupdate RDBMS (SQL/Oracle Sever)

Datenbankarbeiten dürfen nur bei gestoppter Verarbeitung erfolgen.

5.4 Die Datenbanken

5.4.1 Die SQL Server Datenbank

Zur Applikation gehört eine Datenbank, die dazu dient, bestimmte während der Laufzeit anpassbare Werte vorzuhalten. Diese Werte steuern die Verarbeitung oder die Abweisung einer E-Mail z.B. durch Vorgabe der Anzahl an möglichen Attachments auf Basis eines Dokumentenschlüssels.

Diese Datenbank und ihr Inhalt wird durch die AOK vorgegeben und kann jederzeit aktualisiert werden. Ein Restart der Applikation ist dazu nicht notwendig. Die Umgebung muss aber zur Laufzeit erreichbar sein, sonst entsteht ein technischer Fehler, der manuell verarbeitet werden muss.

Benötigt wird hierzu eine Instanz eines Microsoft SQL Servers, die aus der jeweiligen NiFi Umgebung heraus, zu erreichen sein muss.

Die Anwendung besteht aus 5 Tabellen für CKMail:

- Sendungsnummern
- AttachmentRules
- Regelwerk
- Rueckweisungsgruende
- Uebersetzungstabelle

Die Anwendung besteht aus 2 Tabellen für KCFax:

- KC_Sendungsnummer
- KC_Stapelnummer

Sendungsnummern

Mittels dieser Tabelle wird protokolliert und berechnet welche Sendungsnummern bereits vergeben und welche die nächsten sein werden. Diese Tabelle muss nicht vor befüllt werden.

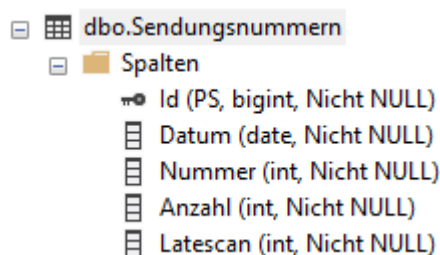


Abbildung 4: Datenbank NiFi Sendungsnummern

Es wird für Epost und Latescan für jeden Tag jeweils neue Nummern angelegt.

Das Datum zeigt das jeweilige Datum an.

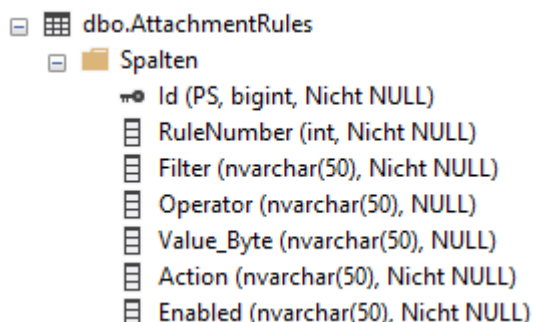
Die Nummer

Die Anzahl von Sendungen

Latescan kann nur 0 oder 1 sein. Eine 1 zeigt einen Latescan an, eine 0 einen EPost Vorgang.

AttachmentRules

Diese Tabelle wird vorausgefüllt und liefert Informationen, welche Anhänge mit welcher Größe verarbeitet oder ignoriert werden.



```

    dbo.AttachmentRules
    Spalten
    Id (PK, bigint, Nicht NULL)
    RuleNumber (int, Nicht NULL)
    Filter (nvarchar(50), Nicht NULL)
    Operator (nvarchar(50), NULL)
    Value_Byte (nvarchar(50), NULL)
    Action (nvarchar(50), Nicht NULL)
    Enabled (nvarchar(50), Nicht NULL)
  
```

Abbildung 5: Datenbank AttachmentRules

Die RuleNumber ist lediglich eine Referenz

Der Filter referenziert die zu bearbeitenden Elemente

Der Operator bewertet den folgenden Value (größer als, kleiner als, etc)

Der Value_Byte gibt einen Grenzwert für den Filter vor

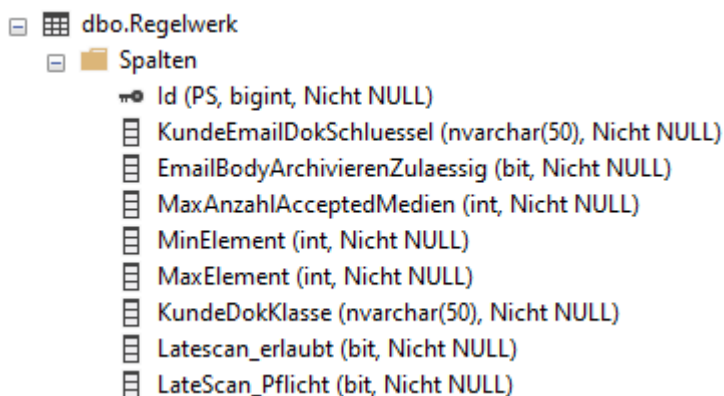
Action ist die Aktion, die im Falle eines zutreffenden Filters durchgeführt werden sollte

Enabled Aktiviert Rule oder setzt sie außer Kraft

Regelwerk

Diese Tabelle wird vorausgefüllt und beinhaltet den „DokSchlüssel“ als Zugriffselement.

Für jeden „DokSchlüssel“ wird festgelegt:



```

    dbo.Regelwerk
    Spalten
    Id (PK, bigint, Nicht NULL)
    KundeEmailDokSchluessel (nvarchar(50), Nicht NULL)
    EmailBodyArchivierenZulaessig (bit, Nicht NULL)
    MaxAnzahlAcceptedMedien (int, Nicht NULL)
    MinElement (int, Nicht NULL)
    MaxElement (int, Nicht NULL)
    KundeDokKlasse (nvarchar(50), Nicht NULL)
    Latescan_erlaubt (bit, Nicht NULL)
    LateScan_Pflicht (bit, Nicht NULL)
  
```

Abbildung 6: Datenbank Regelwerk

Ob eine Body Archivierung zulässig ist

Die maximale Anzahl der akzeptierten Medien

Mindest- und Maximal-Elementzahl des Betreffs

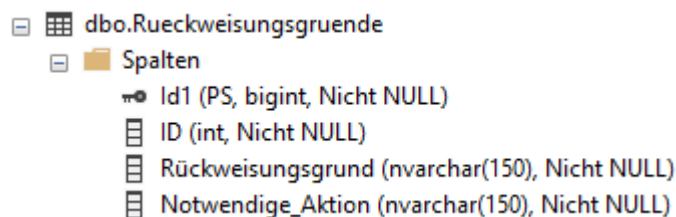
Den Namen der Dokumentenklasse

Ob Latescan erlaubt ist

Ob Latescan verpflichtend ist

Rückweisungsgründe

Diese Tabelle wird vorausgefüllt und liefert die offiziellen Fehlertexte und notwendige Aktionen bei bestimmten Abweisungssituationen.

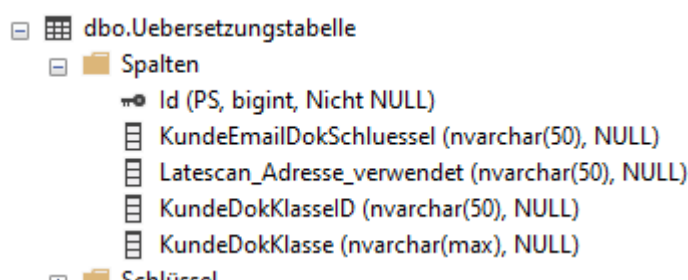


dbo.Rueckweisungsgruende
Spalten
Id1 (PS, bigint, Nicht NULL)
ID (int, Nicht NULL)
Rückweisungsgrund (nvarchar(150), Nicht NULL)
Notwendige_Aktion (nvarchar(150), Nicht NULL)

Abbildung 7: Datenbank Rückweisungsgründe

Übersetzungstabelle

Diese Tabelle wird vorausgefüllt und liefert die YLAMI-Klasse und den Beschreibungstext für einen „DokSchlüssel“. Der DokSchlüssel kommt dabei aus der Betreffzeile.



dbo.Uebersetzungstabelle
Spalten
Id (PS, bigint, Nicht NULL)
KundeEmailDokSchluessel (nvarchar(50), NULL)
Latescan_Adresse_verwendet (nvarchar(50), NULL)
KundeDokKlasseID (nvarchar(50), NULL)
KundeDokKlasse (nvarchar(max), NULL)

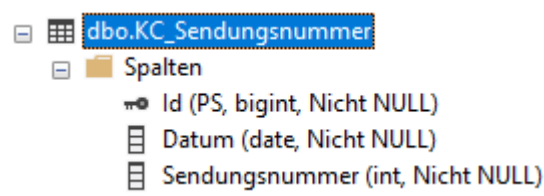
Abbildung 8: Datenbank Übersetzungstabelle

EEmailDokSchlüssel ist die Referenz zu dem anderen Werten.

Alle anderen Werte dieser Tabelle werden als Text in Dateien bzw. Abweisungsmails verwandt.

Die Ansprache erfolgt dabei über eine Connectionstring der aus der Konfigurationsdatei ausgelesen wird und Umgebungsabhängig ist.

KC_Sendungsnummer

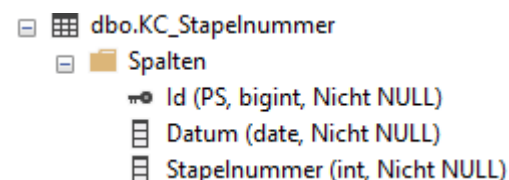


dbo.KC_Sendungsnummer
Spalten
Id (PS, bigint, Nicht NULL)
Datum (date, Nicht NULL)
Sendungsnummer (int, Nicht NULL)

Abbildung 9: KC Sendungsnummer

Diese Tabelle beinhaltet jeweils das Datum und die letzte aktuelle Sendungsnummer. Die Verwendung dieser Tabelle ist wartungsfrei.

KC_Stapelnummer



dbo.KC_Stapelnummer
Spalten
Id (PS, bigint, Nicht NULL)
Datum (date, Nicht NULL)
Stapelnummer (int, Nicht NULL)

Abbildung 10: KC Stapelnummer

Diese Tabelle beinhaltet jeweils das Datum und die letzte aktuelle Stapelnummer. Die Verwendung dieser Tabelle ist wartungsfrei.

5.4.2 Die Oracle Reporting Datenbank

Diese Datenbank ist ausschließlich zu Reportingzwecken vorgesehen. In diese Datenbank werden alle in der Schnittstelle erstellte Stapel in der Struktur Stapel -> Sendung -> Dokument protokolliert und für Auswertungszwecke zur Verfügung gestellt.

Das Füllen der Tabellen erfolgt dabei jeweils beim Stapelabschluss.

Die drei Tabellen repräsentieren dabei jeweils einen Stapel, der mehrere Sendungen enthält, die jeweils mehrere Dokumente enthalten können.

Die Ansprache erfolgt dabei über eine Connectionstring der aus der Konfigurationsdatei ausgelesen wird und Umgebungsabhängig ist.

Die Tabelle Stapel enthält die Informationen für die Stapelebene.

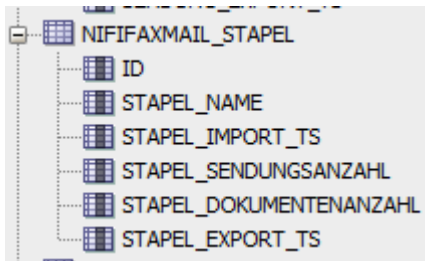


Abbildung 11: Der Reporting Stapel

ID = Eindeutige ID

Stapelname = Eindeutiger Name

Import TS = Zeitpunkt des Imports

Export TS = Zeitpunkt des Exports

Sendungsanzahl = Anzahl der Sendungen im Stapel

Dokumentenanzahl = Anzahl der Dokumente im Stapel

Die Tabelle Sendung enthält die Informationen für die Sendungsebene.

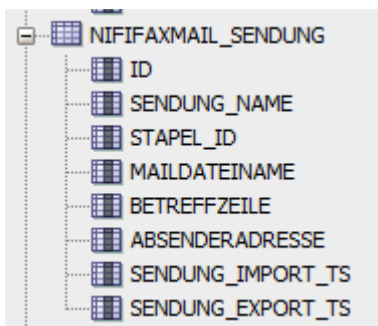


Abbildung 12: Die Reporting Sendung

Sendungsname = Eindeutiger Sendungsname

StapelID = Referenz auf den Stapel

Maildateiname = Name der ursprünglich verarbeiteten Maildatei

Betreffzeile = Die Betreffzeile der E-Mail
 Absenderadresse = Die Absenderadresse der E-Mail
 Import Timestamp = Import TS
 Export Timestamp = Export TS

Die Tabelle Dokument enthält die Information für die Dokumentenebene.

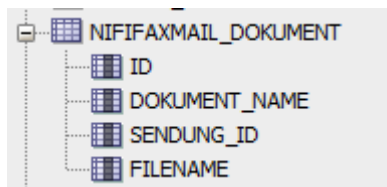


Abbildung 13: Das Reporting Dokument

Dokumentname = Eindeutiger Dokumentenname
 SendungsID = Referenz auf die Sendung
 Filename = Name des ursprünglichen Dokumentes

Die Tabelle Abweisung entgält alle abgewiesenen Email einschließlich ihrer Ursache.

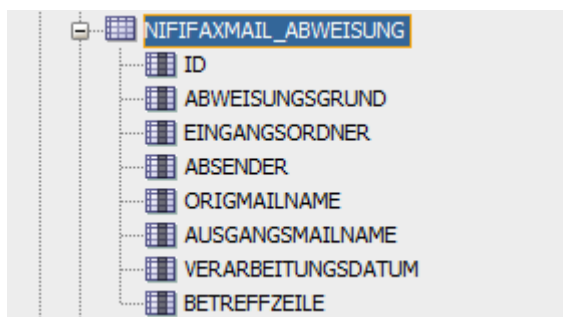


Abbildung 14: Die Tabelle Abweisung

Anweisungsgrund = Grund der Abweisung
 Eingangsordner = EPost oder Latescan
 Absender = Email Adresse des Einsenders
 Originalname = Name der EML Datei aus dem IMAP Kollektor zur Referenz
 Ausgangsmailname = Name der EML Datei die an den Absender gesandt wurde
 Verarbeitungsdatum = Verarbeitungsdatum
 Betreffzeile = Inhalt der Betreffzeile der Originalmail

6 Die Testumgebung

6.1 Der physikalische Server

Die Konfiguration des Windowsserver der Testumgebung:

Servename	DS6D-WW-6B071.AOKSAN.AOK
IP	10.31.107.71
Version Windows Server	Windows Server 2016
Domain	AOKSAN.AOK
User	SAN_SVC_REP
Password	Siehe Password DB
Version Java	openjdk version "17.0.6" 2023-01-17 LTS

Pfad zu Java	C:\Program Files\Microsoft\jdk-17.0.6.10-hotspot\bin\
\$JAVA_HOME	C:\Program Files\Microsoft\jdk-17.0.6.10-hotspot\
Version .Net Core Runtime	.Net Core 5
Version Imagemagick	ImageMagick 7.1.0-4 Q16 x64 2021-07-18
Pfad Imagemagick	C:\NiFIFaxEmail\Base\imagemagick
Zugeordneter DB Server	DS6D-DB-6B226.aoksan.aok
Zugeordnete DB Instanz	NiFi
DB User	SVC_Nifi
Password	Siehe Password DB
Zugeordneter Reporting Server	10.108.209.77:1538/lipmstat.aokt
DB User	REPORTING_WRITE
Password	Siehe Password DB
Zugeordneter Mailserver	SMTPV.AOKSAN.AOK

Tabelle 2: Server in der Testumgebung

6.2 Der NiFi Server

Die Konfiguration des NiFi Servers in der Testumgebung:

Installationspfad	C:\NiFiEnvironment\nifi-1.19.1-bin\nifi-1.19.1
Log Pfad	C:\NiFiEnvironment\nifi-1.19.1-bin\nifi-1.19.1\logs
Config Pfad	C:\NiFiEnvironment\nifi-1.19.1-bin\nifi-1.19.1\conf
Aktive Prozessorgruppe für Test	FaxMAilNew
User	Siehe Password DB
Password	Siehe Password DB

Tabelle 3: Nifi Server in der Testumgebung

Templatefile:



FaxMailTest.xml

6.3 Das Programm

6.3.1 CKMail

Die Konfiguration des CKMail Tools in der Testumgebung:

Installationspfad	C:\NiFIFaxEmail\Bin
Log Pfad	C:\NiFIFaxEmail\Base\protokoll

Tabelle 4: CKMail in der Testumgebung

Die aktuellen Konfigurationsdateien der Testumgebung:



App.config



log4net.config

6.3.2 KCFax

6.3.3 Die Konfiguration des KCFax

Tools in der Testumgebung:

Installationspfad	C:\NiFIFaxEmail\Bin
Log Pfad	C:\NiFIFaxEmail\Base\protokoll

Tabelle 5: CKMail in der Testumgebung

Die aktuellen Konfigurationsdateien der Testumgebung:

6.4 Die Weitergabe

Das Verzeichnis für die Ausgabe und den Abschluss der Stapel in der Testumgebung:

Ausgabeverzeichnis	D:\NIFI\NIFI_build
Abschlusskennung	.rdy

Tabelle 6: Ausgabeschnittstelle CKMail in der Testumgebung

7 Die Produktionsumgebung

7.1 Der physikalische Server

Die Konfiguration des Windowsserver der Produktionsumgebung:

Servename	DS6D-WW-6B072.AOKSAN.AOK
IP	10.31.107.72
Version Windows Server	Windows Server 2016
Domain	AOKSAN.AOK
User	SAN_SVC_REP
Password	Siehe Password DB
Version Java	openjdk version "17.0.6" 2023-01-17 LTS
Pfad zu Java	C:\Program Files\Microsoft\jdk-17.0.6.10-hotspot\bin\
\$JAVA_HOME	C:\Program Files\Microsoft\jdk-17.0.6.10-hotspot\
Version .Net Core Runtime	.Net Core 5
Version Imagemagick	ImageMagick 7.1.0-4 Q16 x64 2021-07-18
Pfad Imagemagick	C:\NiFIFaxEmail\Base\imagemagick
Zugeordneter DB Server	DS6D-DB-6B226.aoksan.aok
Zugeordnete DB Instanz	NiFi_Prod
DB User	SVC_Nifi
Password	Siehe Password DB
Zugeordneter Reporting Server	10.108.41.53:1531/lipmstap.aokp

	Handbuch
	Fax/E-Mail Schnittstelle

DB User	REPORTING_WRITE
Password	Siehe Password DB
Zugeordneter Mailserver	SMTPV.AOKSAN.AOK

Tabelle 7: Server in der Produktionsumgebung

7.2 Der NiFi Server

Die Konfiguration des NiFi Servers in der Testumgebung:

Installationspfad	C:\NifiFLProd\nifi-1.19.1
Log Pfad	C:\NifiFLProd\nifi-1.19.1\logs
Config Pfad	C:\NifiFLProd\nifi-1.19.1\conf
Aktive Prozessorgruppe für Test	Fax EMail Schnittstelle
User	Siehe Password DB
Password	Siehe Password DB

Tabelle 8: Nifi Server in der Produktionsumgebung

NiFi Templatefile für die Prozessgruppe:



FaxMailProd.xml

7.3 Das Programm

7.3.1 CKMail

Die Konfiguration des CKMail Tools in der Produktionsumgebung:

Installationspfad	C:\NifiFaxEmail\Bin
Log Pfad	C:\NifiFaxEmail\Base\protokoll\protokoll.log

Tabelle 9: CKMail in der Produktionsumgebung

Die aktuelle Konfigurationsdatei der Produktionsumgebung:



App.config



log4net.config

7.3.2 KCFax

Die Konfiguration des KCFax Tools in der Produktionsumgebung:

Installationspfad	C:\NifiFaxEmail\Bin
Log Pfad	C:\NifiFaxEmail\Base\protokoll\protokoll.log

Tabelle 10: KCFax in der Produktionsumgebung

Die aktuelle Konfigurationsdatei der Produktionsumgebung:

7.4 Die Weitergabe

Das Verzeichnis für die Ausgabe und den Abschluss der Stapel in der Produktionsumgebung:

Ausgabeverzeichnis	D:\NIFIDATA\Build
Abschlusskennung	.rdy

Tabelle 11: : Ausgabeschnittstelle CKMail in der Produktionsumgebung

8 Der Change Prozess

8.1 Der Anpassungsprozess für Änderungen der Datenbanktabellen

8.1.1 Anlegung einer neuen Dokuklasse

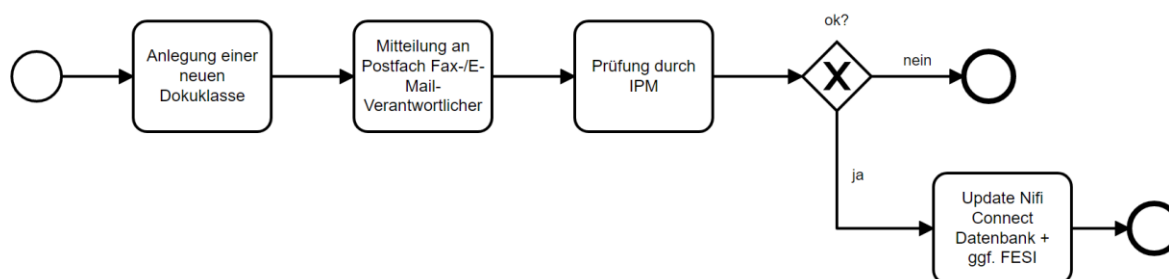


Abbildung 15: Der Change Prozess

Wird eine neue Dokuklasse angelegt, wird dieses per E-Mail an den Verantwortlichen der Fax-/E-Mail-Schnittstelle kommuniziert. Die Anforderung wird geprüft. Entspricht die Anforderung den formalen Voraussetzungen, so wird die Datenbank der Fax-/E-Mail-Schnittstelle durch das Team Inputmanagement geupdated. Die Aktualisierung von FESI erfolgt separat.

8.1.2 Löschen einer Dokuklasse

Der Auftrag zum Löschen in der Fax-/E-Mail-Schnittstelle erfolgt ebenfalls per E-Mail.

8.1.3 Änderung formeller Prüfungen bestehender Dokuklassen

Unter bestimmten Voraussetzungen können einzelne formelle Prüfungen angepasst werden, z.B. die minimale und maximale Anzahl von Elementen in der Betreffzeile. Dies trifft dann zu, wenn in einem Bereich für bestimmte Dokumentklassen die Gebührenposition obligatorisch und für andere optional ist.

Allgemein

Für eePV-Dokuklassen gelten folgende Standardregeln:

Dok-schlüssel	Body	maxMedien	minElement	maxElement	DokKlasse	LateScan erlaubt	LateScan Pflicht
Beispiel	True	99	2	2	Beispiel	True	False

Tabelle 12: eePV Dokuklassen

Die Anzahl der Elemente in der Betreffzeile können grundsätzlich zwischen 1 und 4 variieren (Dokschlüssel, KV-Nummer, Fall-/Fallbündelnummer, Gebührenposition) und ist für jede Dokuklasse individuell zu bewerten.

Für APD-Dokuklassen (technische DOkuklassen) gelten folgende Standardregeln:

Dok-schlüssel	Body	maxMedien	minElement	maxElement	DokKlasse	LateScan erlaubt	LateScan Pflicht
Beispiel	False	1	2	2	Technischer Dummy	False	False

Tabelle 13: APD Dokuklassen

Es ist zu beachten, dass für APD-Dokuklassen keine Body-Archivierung zulässig ist und die Anzahl der maximalen E-Mail-Anhänge auf 1 beschränkt ist.

8.1.4 Technische Aktivitäten:

Für alle Tabellen gilt, dass bei Änderungen des Inhaltes zu prüfen ist, ob eine Programmanpassung notwendig ist. Eine generische Aussage kann hier nicht gemacht werden, da die Bedeutungen von beliebigen Änderungen nicht prognostiziert werden können.

Dies gilt im Besonderen für die Tabelle AttachmentRules, da die Bedeutung einer neuen ACTION nicht antizipiert werden kann.

Bei neuen Doku-Klassen ist ein Löschen oder hinzufügen von neuen Klassen problemlos möglich, sofern die Schlüsselkonditionen eingehalten werden (DokSchlüssel eindeutig).

Das Update kann manuell oder per Skript erfolgen. Eine Referenz der durchgeführten Änderungen sollte dokumentiert werden.

Technisch gesehen ist ein Datenbankupdate im laufenden Betrieb möglich, aber nicht ratsam.

9 Referenzen

9.1 Der Anpassungsprozess für Änderungen der Datenbanktabellen

SQL-Server: DS6D-DB-6B226.aoksan.aok
 Datenbankinstanz: NiFi
 User: SVC_NiFi
 Password:

9.2 Datenbanklogin Prod:

SQL-Server: DS6D-DB-6B226.aoksan.aok
 Datenbankinstanz: NiFi_Prod
 User: SVC_NiFi
 Password:

9.3 Erstellung eines verschlüsselten Passwords:

Das Programm enthält eine Funktion zum verschlüsseln und entschlüsseln von Passwörtern.

Der Aufruf erfolgt wie folgt in der Kommandozeile:

```
># CKMail.exe c Passwort
```

Dass Programm zeigt das verschlüsselte Passwort an.
Zum Entschlüsseln kann man die Funktion d verwenden:

```
># CKMail.exe d EncryptedPassword
```

Dass Programm zeigt das originale Passwort an.

9.4 Postfächer:

Umgebung	Adresse	Berechtigungen	Postfach
Produktion	prod.epost@san.aok.de	AOKSAN\SAN_SVC_Nifi	LF00739
Test	qa.epost@aoksanint.de	AOKSANINT\SAN_SVC_Nifi	TLF00017
Produktion	prod.latescan@san.aok.de	AOKSAN\SAN_SVC_Nifi	LF00740
Test	qa.latescan@aoksanint.de	AOKSANINT\SAN_SVC_Nifi	TLF00016
Produktion KCFax			
Test KCFax			

Abbildung 16: Postfächer

<https://outlook.aoksanint.aok/>

<https://outlook.aoksan.aok/>



The image shows the Outlook login interface. At the top, there is the Outlook logo, which consists of a blue square with a white 'O' and a white envelope icon, followed by the word 'Outlook' in a large, blue, sans-serif font. Below the logo, there are two input fields. The first is labeled 'Domäne\Benutzername:' and contains the text 'aoksan\lf00740'. The second is labeled 'Kennwort:' and is empty. Below the input fields, there is a blue circular button with a white right-pointing arrow and the text 'Anmelden' in a blue, sans-serif font.

Abbildung 17: Outlook Login

9.5 Das Programm CKMail.exe im Detail

Allgemein: Das Programm nutzt das DoTnet Core Framework 5.0 und das Entity Framework Core 5 zur Datenverarbeitung.

Für Loggingzwecke wird das Log4Net Framework angewandt.

Gegebenenfalls erzeugte E-Mails werden durch das MimeKit und das MailKit erzeugt, bzw. versandt. Die HTML-Unterstützung erfolgt durch das HtmlAgilityPack.

Alle verwendeten externen Bibliotheken:

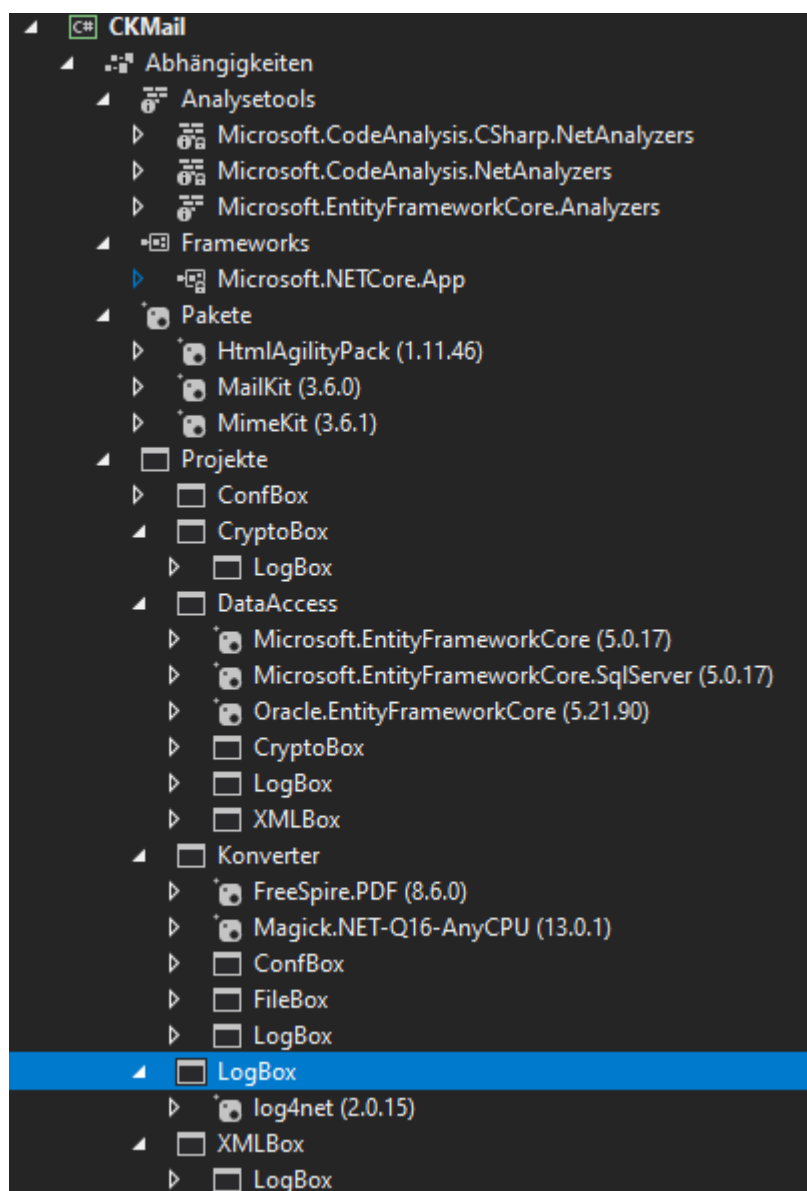


Abbildung 18: Bibliotheken in CKMail

Damit ist die obige Liste die Referenz der benutzten Open Source Frameworks zur Erstellung des Programms.

Die Dateien im Verzeichnis Daten sind mittels EF 5 Framework aus der Datenbank „reverse engineered“ und werden für die Kommunikation zwischen Programm und DB benutzt.
Alle anderen Klassen dienen der Datenverarbeitung.

Die Klassen:

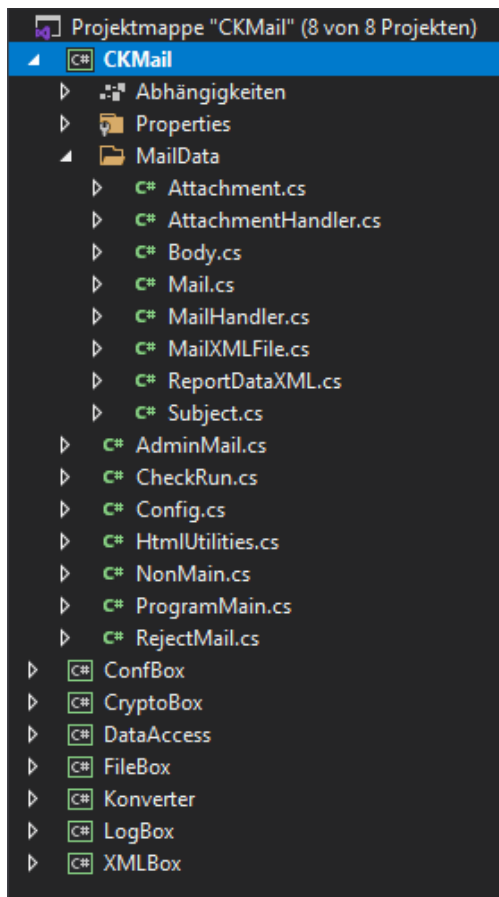


Abbildung 19: Klassenstruktur CKMail

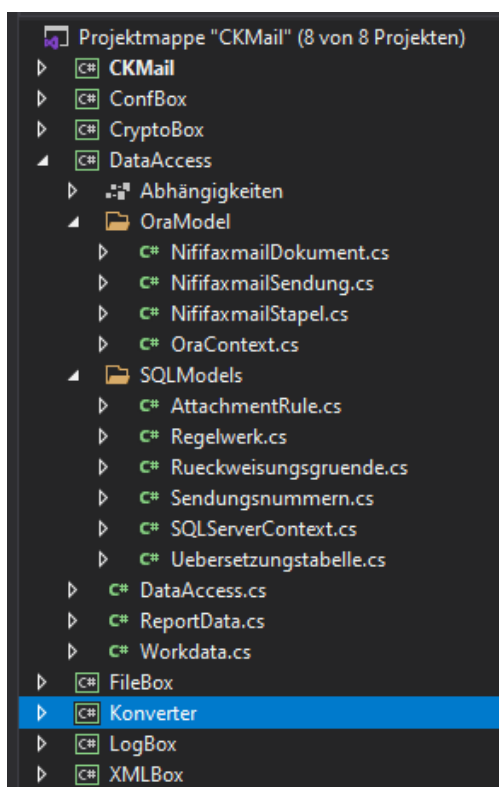


Abbildung 20: Klassenstruktur DataAccess

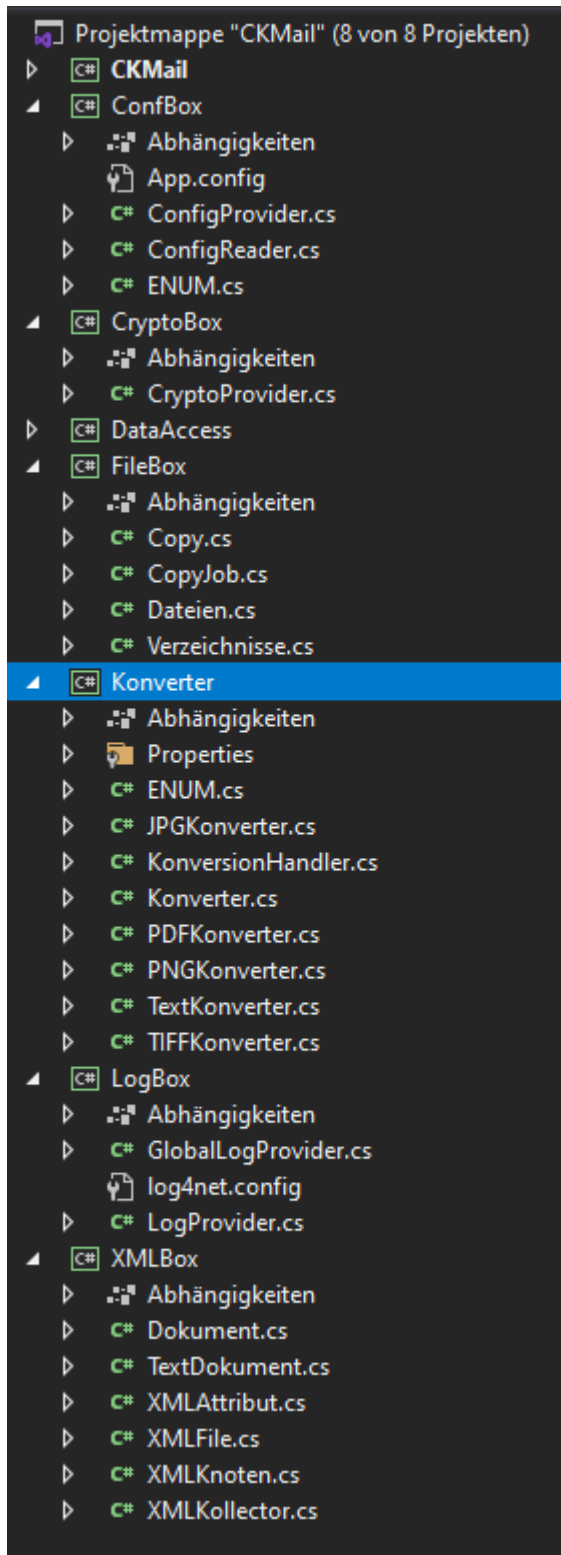


Abbildung 21: Klassenstruktur Hilfsbibliotheken

9.5.1 Klasse Programm (ProgramMain.cs und NonMain.cs)

Diese Klasse enthält die Mainfunktion und steuert den Programmablauf.

Diese Funktion ist der Einsprungpunkt der Verarbeitung. Sie bekommt entweder eine Email Datei oder die „check“ - Datei übergeben.

Es wird zunächst der Übergabeparameter eingelesen und kontrolliert, ob eine Datei mit dem übergebenen Namen existiert und liest dann die Konfiguration ein und überprüft die einzelnen Elemente der Konfiguration.

```
ProcessParameter(args)
```

```
MyConfig = new();
```

Danach wird die Verbindung zu den beiden Datenbanken überprüft:

```
MyWorkData = new(MyConfig.DbUser, MyConfig.DbPassword,
MyConfig.DbConnectionString);
MyReportData = new(MyConfig.OraReport_Connector);
```

In der Folge wird das evtl. vorhandene Arbeitsverzeichnis gelöscht und der Checkrun, also der Stapelabschluss ausgeführt.

```
CheckRun.PrepEnv()
```

Und

```
CheckRun.ProcessCheckRun()
```

Dieser überprüft ob es 24h verzögerte E-Mails gibt (Zugewinn) und benennt diese entsprechend um und kopiert sie zurück in den Eingangsordner zur endgültigen Verarbeitung.

Danach werden alle Stapel die die Grüßen- oder Altersbedingungen erfüllen abgeschlossen, also mit der Endung versehen, die die Weiterverarbeitung auslöst,

Ist eine „check“ Datei übergeben worden so wir das Programm hier beendet.

```
if (Path.GetFileNameWithoutExtension(Mailfile) == "check")
{
    BeendeProgramm(ErrorCode.Success);
}
```

Wurde eine E-Mail übergeben, so erfolgt nun die normale E-Mailverarbeitung. Diese hat grundsätzlich die Schritte

- Einlesen

-

```
MyMailHandler = new(Mailfile);
MyMailHandler.ReadMail()
```

- Auswerten

```
MyMailHandler.ProcessMail()
```

- Gegebenenfalls eine Zugewinnmail in den Delayordner kopieren

```
if (MyMailHandler.MyMail.DelayedMail)
{
    MailHandler.CopyToDelayFolder()
}
```

- Persistieren

```
MyMailHandler.WriteMail()
```

- Konvertieren
`myKonverter.KonvertData()`
- Ausgabe in das Stapelverzeichnis
`MailHandler.CopySendungToStapel()`
- Löschen des Arbeitsverzeichnisse als Bestandteil der Funktion BeendeProgramm()
`CleanUp()`

Der abschließende Schritt hat den Vorteil, dass es bei Konvertierungsfehler nicht wie früher zu Artefakten kommen kann, die eine manuelle Intervention erfordern.

Allgemein gilt:

Im Falle einer Fachlichen Abweisung erfolgt eine Rückmail an den Einsender durch Instanziierung der Klasse RejectMail.

Im Falle eines technischen Fehlers erfolgt eine Warnmail an den in der Konfiguration hinterlegten Admin-Mailaccount durch Instanziierung der Klasse AdminMail.

Sollte der Mailrelayserver einmal nicht verfügbar sein, müssen andere Tools wie z.B. Zabbix das Monitoring übernehmen. Sinnvolle Parameter sind hier der Rückgabewert -> NiFi Fehlerdatei oder die Kennung „ERROR“ in der Logdatei. Diese Kennung weist immer auf einen technischen Fehler hin und sollte überprüft werden.

9.5.2 Die Klassen zur Verarbeitung der E-Mail:

Eine E-Mail besteht aus einem Body, einem Subject, einer oder mehrere Adressaten und möglicherweise mehreren Attachments.

Für jeden dieser Bereiche gibt es eine gleichnamige Verarbeitungsklasse, in der jeweils alle relevanten Schritte abgehandelt werden. Sind mehrere Elemente vorhanden, so gibt es auch eine „Handler“ Klasse“. Diese sorgt für die wiederholte Verarbeitung der einzelnen Elemente als verknüpfte Liste.

Zwei besondere Klasse sind dabei die Klassen „MailXMLFile“ und „ReportDataXML“. Erstere schreibt das XML File für die Weiterverarbeitung, zweitere enthält alle Daten für das Reporting beim Stapelabschluss und wird nach Stapelabschluss gelöscht.

9.5.3 Die Klasse CheckRun:

Die Klasse kopiert verzögerte E-Mails zurück in den Eingangsordner, check das Alter und die Stapelgröße und bereitet die Stapel gegebenenfalls auf die Weiterverarbeitung vor. Dazu wird der Stapel in die Reportingdatenbank eingetragen und eine neue Stapelkennung gesetzt. Die Reporting XML Datei wird dabei gelöscht.

9.5.4 Die Klassen in der Bibliothek DataAccess:

Diese Dateien sind mittels Entity Framework Manager generiert und stellen die Repräsentation und Zugriffe der beiden Datenbanken dar. Sie erlauben den Zugriff auf die Datenbank („Context“ Klasse).

Die anderen Klassen stellen jeweils die Repräsentation einer Tabelle dar. (Details: [Entity Framework Manager](#)).

Die Klasse Reportdata stellt die Funktionen für den Zugriff auf die Reportingdatenbank dar, die Klasse WorkData beinhaltet die Funktionen für den Zugriff auf die MS-SQL Datenbank.

9.5.5 Die Konverter:

Die Bibliothek Konverter enthält Klassen zur Konvertierung von Datentypen. Hier sind folgende Klassen (bzw. Datentypen) vorgesehen:

Der KonversionHandler liest alle Dokumentarten ein und steuert die jeweilige Verarbeitung einzeln an.

PDF Dokumente:

Steuert die Konvertierung der PDF Dokumente. Hier werden diese in strukturierte PDF Dokumente konvertiert.

Des Weiteren gibt es eine statische Funktion zur Durchführung von Vorabtests um die Verarbeitungsstabilität zu erhöhen. Diese Tests werden vor der Konvertierung durchgeführt.

PNG Images:

Steuert die Konvertierung der PNG Dokumente. Hier werden diese in strukturierte PNG Dokumente konvertiert.

JPG Images:

Steuert die Konvertierung der JPG Dokumente. Hier werden diese in strukturierte JPG Dokumente konvertiert.

TIF/TIFF Images:

Steuert die Konvertierung der TIF/TIFF Dokumente. Hier werden diese in strukturierte TIF Dokumente konvertiert.

Text Dokumente:

Steuert die Konvertierung der Text Dokumente. Hier werden diese in TIF Dokumente konvertiert.

9.5.6 Die Bibliotheken:

Die anderen Bibliotheken sind kleine Hilfsprogramme, die spezialisierte Funktionen abbilden.

ConfBox:

Erlaubt das automatisierte Auslesen von Konfigurationsdateien.

CryptoBox:

Erlaubt das verschlüsselte Ablegen von Konfigurationsdaten.

FileBox:

Stellt Funktionen zur sicheren Arbeit mit Dateien zur Verfügung.

LogBox:

Erlaubt das einfache Einrichten von Loggern.

XMLBox:

Stellt Funktionen zur Bearbeitung von XML Dateien zur Verfügung.

9.5.7 Die KonfigDatei App.Config

[Die Konfiguration:](#)

9.5.8 Die Klassenreferenzen

Die gegenwärtige beschriebene Version ist:

Release 2.0.0

Dabei steht die 1 Zahl für eine vollwertige Version.

Dabei steht die 2 Zahl für ein Featureupdate.

Dabei steht die 3 Zahl für ein Bugfix

9.5.9 Das Programm ImageMagick

<https://imagemagick.org/script/download.php#windows>

9.5.10 Das Programm Repository CKMail

https://VRS2D-SO-1E097.aoksan.aok/svn/NIFI_CONNECT

9.5.11 Das Programm Repository KCFax

https://VRS2D-SO-1E097.aoksan.aok/svn/NIFI_FAX_KC

9.6 Rückweisungsgründe

In dieser Version sind folgende Rückweisungsgründe definiert und/oder in Verwendung.
Ein Rückweisungsgrund setzt sich aus zwei Elementen zusammen.

1. Ein kurzer und sprechender Name:
z.B.: NoDocument
2. Eine zugehörige ID (positiver ganzzahliger Integer Wert).
z.B. 1

Dieser Wert ist ein Schlüssel, mit dem aus der Arbeitsdatenbank ein Text extrahiert wird, der dann in der Rückweisungsmail übermittelt wird.

9.6.1 Referenz:

```

KeineRueckweisung = 0,
NoDocument = 1,           // Kein zu archivierendes Dokument
InvalidFormat = 2,        // Ungültiges Dateiformat
InvalidSubject = 3,       // Ungültiger Betreff
NoSuchDocuKlasse = 4,     // Unbekannte Dokumentenklasse
WrongSubjectFieldCount = 5, // Falsche Feldanzahl im Betreff
ForbiddenBodyArchiving = 6, // Bodyarchivierung verboten
WrongAttachmentCount = 7, // Falsche Anzahl Anhänge
BothMailAddresses = 8,    // an beide Adressen versendet
NoLateScanAllowed = 9,    // Latescan ist nicht erlaubt
WrongMailAddress = 10,    // Falsche Mailadresse
LateScanRequired = 11,    // Latescan ist verpflichtend
RegelwerkDisabled = 12,   // Regelwerk ist ausgeschaltet(enabled = false)
TiffToBig = 13,           //Tiff ist größer als 32000x24000
PDFEncrypted = 14,        //Das PDF Dokument ist encrypted
CorruptedPDF = 15,        //Bittiefe der Ergebnisdatei
SpecialCharacter = 16,    //Sonderzeichen im Attachment
PDFMaxPageCount = 17,    //Maximale Anzahl der Seiten eines zu konvertierenden
PDF Dokumentes

```

9.6.2 Erklärung:

Keine Rückweisung:

Dies ist die Defaultoption, die im Programmfluss nach einem Test zurückgegeben wird, wenn keine Aussteuerung erfolgen soll.

NoDocument:

Rückgabeoption für den Test ob ein Attachment und oder ein Body veraktet werden soll. Liegt kein Element zur Veraktung an, ist dies die Rückgabeoption die

InvalidFormat:

Es wurde ein Attachment mit einem nicht erlaubten Dateiformat gefunden. Erlaubte Formate stehen in der AttachmentRules Tabelle der Arbeitsdatenbank. Dabei wird aber nur die Endung und nicht der wirkliche Typ überprüft -> Dies kann eine Fehlerquelle sein, da eine Konvertierung aus diesem Grund fehlschlagen kann.

InvalidSubject:

Die Betreffzeile der Email konnte nicht erfolgreich ausgewertet werden.

NoSuchDocuKlasse:

Die ermittelte Dokumentenklasse (aus der Betreffzeile) befindet sich nicht in der Regelwerk Tabelle in der Arbeitsdatenbank.

WrongSubjectFieldCount:

Die Betreffzeile ist weisst zu viele oder zu wenige Elemente auf. Die Anzahl der erlaubten Elemente der Betreffzeile ist in der Regelwerttabelle in der Arbeitsdatenbank festgelegt. (MinElement und MaxElement)

ForbiddenBodyArchiving:

Für die ermittelte Dokumentenklasse ist eine Archivierung des Mailbodys nicht zulässig, es wurde aber die Option M_ (Archivierung des Bodys angefordert) in der Betreffzeile übergeben.

WrongAttachmentCount:

Die Anzahl der mitgelieferten Attachments ist nicht erlaubt. Die Anzahl der erlaubten Attachments ist in der Tabelle Regelwerk unter „MaxAnzahlAcceptedMedien“ festgelegt.

BothMailAddresses:

Zur Unterscheidung der Verarbeitungswege ist eine eindeutige Ermittlung des Eingangspostfaches notwendig. Sollte die Mail an beide Postfächer gesendet worden sein, so ist keine Zuordnung möglich und es erfolgt eine Abweisung mit dieser Kennung.

NoLateScanAllowed:

Die E-Mail wurde an das Latescan Postfach gesandt. Es ist aber keine Latescan-Verarbeitung erlaubt. Die Information ist in der Tabelle Regelwerk in der Spalte „LateScan_erlaubt“ dokumentiert.

Erlaubt = 1

Nichte erlaubt = 0

WrongMailAddress:

Es wurde eine E-Mail eingesteuert, die weder e-Post noch LateScan repräsentiert. Dies kann in der gegenwärtigen Konfiguration nur manuell passieren.

LateScanRequired:

Die E-Mail wurde über das e-Post Verfahren eingesteuert. Die Dokumentenklasse erfordert aber zwingend eine LateScan-Einsteuerung.

Die Information ist in der Tabelle Regelwerk in der Spalte „LateScan_Pflicht“ dokumentiert.

Pflicht = 1

Nicht Pflicht = 0

RegelwerkDisabled:

Die Verarbeitung einer Dokumentklasse kann in einer Umgebung an oder ausgeschaltet werden. Dies geschieht über den Eintrag Enabled in der Tabelle Regelwerk.

Verarbeitung aktiv = 1

Verarbeitung nicht aktiv = 0

TiffToBig:

Dies ist aus der Version 1.0 übernommen, da früher PDF in Tiff gewandelt wurden. Diese Prüfung ist nicht mehr aktiv. Sie sollte verhindern, dass zu große Tiff-Dokumente weitergeleitet wurden.

PDFEncrypted:

Verschlüsselte bzw. Password geschützte Dokumente können nicht verarbeitet werden. Wenn ein solches gefunden wird, erfolgt eine Abweisung mit dieser Kennung.

CorruptedPDF:

Diese als PDF bezeichnete Dokument konnte nicht von Spire eingelesen werden und ist vermutlich korrupt. Es erfolgt eine Abweisung.

Grundsätzlich erfolgt die Prüfung durch Einlesen in eine Bibliothek. Diese ist unter Umständen nicht kompatibel zu neuesten PDF Spezifikationen von Adobe und sollte im Problemfall im Visual Studio um NuGet Bereich angepasst werden.

SpecialCharacter:

Nicht mehr in Benutzung. Diente in der alten Version zur Abweisung, wenn besondere Zeichen im Attachment zu finden waren. Mit der neuen Version (ab 2.0.0) obsolete.

PDFMaxPageCount:

Dient zur Abweisung von PDF-Dokumenten mit einer zu großen Seitenzahl. Die Konvertierung von PDF-Dokumenten bedarf Festplattenplatz auf dem Laufwerk C:\.

Die Zahl ist so anzupassen das in der jeweiligen Umgebung genug Platz vorhanden ist. (ca. 100MB pro Seite)

9.7 Die Unterschiede von KCFax zum Programm KCFax.exe im Detail

Folgende Unterschiede bestehen zwischen der CKMai und der KCFax Version

- Die Konfigurationsdatei log4net.conf ist identisch zu CKMail. Es sollte hier nur ein anderer Ablagepfad vorgegeben werden.
- Die Konfigurationsdatei App.Config ist ein Subset der CKMail Datei, da einige Konfigurationsoptionen zur Konvertierung und Dateiverzögerung nicht benötigt werden.
- Es werden nur PDF-Dateien verarbeitet, alle andere Dateitypen werden abgewiesen.
- Es erfolgt keine fachliche Aussteuerung. Alle Fehlerfälle führen zu einer Admin-Mail und einer Fehleraussteuerung