



# Benutzerhandbuch MediPort Communicator für Leistungserbringer und Spitäler

BETRIEB / AUSFÜHRUNG: HANDBÜCHER

Version 1.20.0, 11.10.2016

Dokument-Empfänger	Extern
Dokument-Eigner	MediData AG
Status	Genehmigt
Ersteller	MediData AG
Klassifizierung	ÖFFENTLICH
Copyright	© 2016 MediData AG, Root Längenbold

## Änderungsgeschichte Dokumentation

Rev. Version	Status	Änderungsgrund Beschreibung der Änderung	Änderungs- datum	Prüf- datum	Erstell- er	Genehmi- gung
1.0	Erstellt	Erstfassung	06.10.2004	06.10.2004	sum	PM
1.1	Erstellt	Überarbeitung für MPCommunicator Version 1.2.0	19.08.2003	19.08.2003	sum	PM
1.2	Erstellt	Überarbeitung für MPCommunicator Version 1.3.0	05.04.2004	05.04.2004	sum	PM
1.3	Erstellt	Überarbeitung für MPCommunicator Version 1.4.0	28.09.2004	28.09.2004	sum	PM
1.4	Erstellt	Überarbeitung und Neustrukturierung der Kapitel	31.10.2004	31.10.2004	sum	PM
1.5	Erstellt	Überarbeitung für MPCommunicator Version 1.5.0	28.04.2005	28.04.2005	sum	PM
1.6	Erstellt	Überarbeitung für MPCommunicator Version 1.6.0	01.07.2005	01.07.2005	sum	PM
1.6.1	Erstellt	Anpassung XML Schema GetDocumentList	10.08.2005	10.08.2005	dba	PM
1.6.2	Erstellt	Erweiterung der Mandantenfähigkeit auf 100	28.09.2005	28.09.2005	scr	PM
1.7	Erstellt	Überarbeitung für MPCommunicator Version 1.7.0	21.11.2005	21.11.2005	sca	PM
1.8	Erstellt	Überarbeitung für MPCommunicator Version 1.8.0	23.01.2006	23.01.2006	sca	PM
1.9	Erstellt	Überarbeitung für MPCommunicator Version 1.9.0	10.08.2007	10.08.2007	sca	PM
1.9.1	Erstellt	Kompatibilität für Windows Vista hinzugefügt	22.08.2008	22.08.2008	sca	PM
1.9.2	Erstellt	Überarbeitung für MPCommunicator Version 1.9.2	12.08.2010	12.08.2010	sca	PM
1.9.3	Erstellt	Überarbeitung für MPCommunicator Version 1.9.3	17.01.2011	17.01.2011	sca	PM
1.10.0	Erstellt	Überarbeitung für MPCommunicator Version 1.10.0	24.03.2011	24.03.2011	sca	PM
1.11.0	Erstellt	Überarbeitung für MPCommunicator Version 1.11.0	29.11.2011	29.11.2011	sca	PM
1.12.0	Erstellt	Überarbeitung für MPCommunicator Version 1.12.0	30.01.2012	30.01.2012	sca	PM
1.13.0	Erstellt	Überarbeitung für MPCommunicator Version 1.13.0	23.03.2012	23.03.2012	sca	PM
1.14.0	Erstellt	Überarbeitung für MPCommunicator Version 1.14.0	01.03.2013	01.03.2013	sca	PM
1.14.1	Erstellt	Überarbeitung Layout, Grafiken etc.	09.04.2013	09.04.2013	PM	PM
1.15.0	Erstellt	Überarbeitung für	16.09.2013	16.09.2013	sca	PM

Rev. Version	Status	Änderungsgrund Beschreibung der Änderung	Änderungs- datum	Prüf- datum	Erstell- er	Genehmi- gung
		MPCcommunicator Version 1.15.0				
1.16.0	Erstellt	Überarbeitung für MPCcommunicator Version 1.16.0	12.11.2013	12.11.2013	sca	PM
1.17.0	Erstellt	Überarbeitung für MPCcommunicator Version 1.17.0	14.01.2014	14.01.2014	sca	PM
1.18.0	Erstellt	Überarbeitung für MPCcommunicator Version 1.18.0	14.03.2014	14.03.2014	sca	PM
1.19.0	Erstellt	Überarbeitung für MPCcommunicator Version 1.19.0	25.08.201	25.08.2014	sca	PM
		Update cron-Beschreibung getdocstate im Kapitel 5.3.1	19.01.2015		sty	
1.19	Erstellt	Überarbeitung für MPCcommunicator Version 1.19	19.01.2015		dic	
		neuer Parameter im Kapitel 5.2.2	22.01.2015		sty	
1.19	Erstellt	Überarbeitung Kapitel 2.3.1 Überarbeitung Kapitel 3.1.1 Neues Kapitel 3.2.2 Neues Kapitel 3.2.3 Überarbeitung Kapitel 5.2.4 Überarbeitung Kapitel 5.2.5 Überarbeitung Kapitel 6 Neus Kapitel: 9.2 Neues Kapitel 13.1.2 Neues Kapitel 13.1.3 Überarbeitung Kapitel 15	23.01.2015		dic	
1.19	Erstellt	Überarbeitung Kapitel 12.1	23.03.2015		dic	
1.20	Erstellt	Überarbeitung für MPCcommunicator Version 1.20.0 MacOS 10.12, Windows Server Version 2016 Kompatibilität	28.07.2016	11.10.2016	sca	PM

## Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>6</b>
1.1 Was ist MediPort	6
1.2 Allgemein	7
1.3 Positionierung	7
1.4 Workflow TP und TG	8
<b>2. Grundlagen</b>	<b>10</b>
2.1 Ziele	10
2.2 Architektur	10
2.3 Systemanforderungen	10
<b>3. Funktionen</b>	<b>12</b>
3.1 Daten senden	12
3.2 Daten empfangen	19
3.3 Partnerverzeichnis	28
3.4 Statusverfolgung	29
3.5 Senden ohne Sende-Steuerdateien	32
3.6 Archivierung	36
3.7 Generische Funktionen	38
3.8 Multi User Funktionalität	39
3.9 Sicherheitsfunktionen / Zertifikate	40
3.10 Prozesssteuerung	41
3.11 Protokollierung	42
<b>4. Installation</b>	<b>43</b>
4.1 Allgemein	43
4.2 Windows spezifisch	45
4.3 Linux / Unix spezifisch	46
<b>5. Konfiguration</b>	<b>48</b>
5.1 Minimale Konfiguration	48
5.2 Optionale Konfiguration	49
5.3 Advanced Konfiguration	57
5.4 Konfiguration des Loggers	59
5.5 Test der Konfiguration	61
<b>6. Update</b>	<b>63</b>
6.1 Update	63
<b>7. Betrieb</b>	<b>64</b>
7.1 Allgemein	64
7.2 Windows spezifisch	66

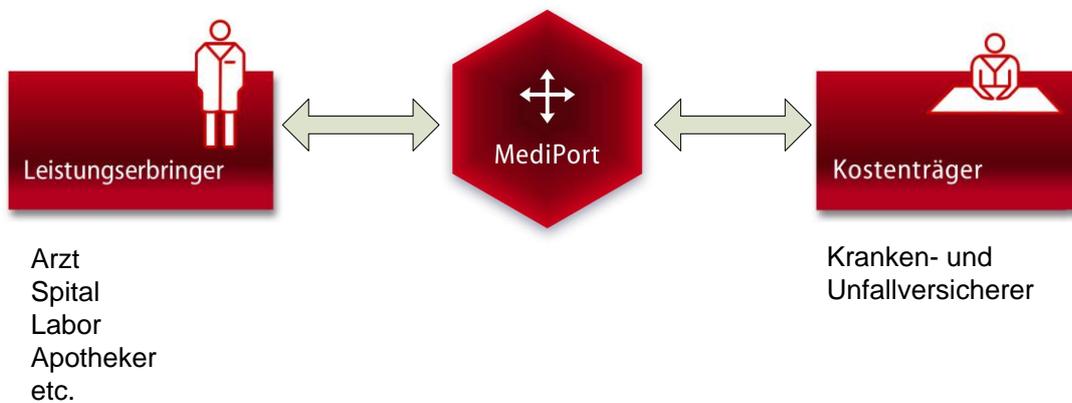
<b>8. Unterhalt</b>	<b>68</b>
8.1 Zertifikatswechsel	68
8.2 Applikation	68
8.3 Datenbank	68
<b>9. Deinstallation</b>	<b>69</b>
9.1 Allgemein	69
9.2 Deinstallation unter Mac OS X	69
<b>10. Problembehandlung</b>	<b>69</b>
10.1 Problem mit Microsoft Proxy	69
<b>11. Anhang A: Dateinamen der empfangenen Dokumente</b>	<b>70</b>
<b>12. Anhang B: Log Meldungen (Auszug)</b>	<b>76</b>
12.1 Meldungen vom Log Level INFO (I0000 – I9999)	76
12.2 Meldungen vom Log-Level WARN (W0000 – W9999)	77
12.3 Meldungen vom Log-Level ERROR (E0000 – E9999)	79
12.4 Meldungen vom Log-Level FATAL (F0000 – F9999)	84
<b>13. Anhang C: Konfiguration und Logging</b>	<b>87</b>
13.1 MPC Konfigurationsdateien	87
13.2 MPCCommunicator Logging	99
<b>14. Anhang D: Scheduler Konfiguration</b>	<b>103</b>
14.1 Cron Expressions	103
<b>15. Anhang E: Release Notes</b>	<b>105</b>

## 1. Einleitung

### 1.1 Was ist MediPort

MediPort ist eine Internet- basierende Plattform für das Gesundheitswesen, welche es den Leistungserbringern erlaubt, elektronisch mit den Kostenträgern abzurechnen. Es sind folgende Parteien in den Prozess involviert:

- Leistungserbringer (Sender)
- Intermediär (MediPort) (Vermittler)
- Kostenträger (Empfänger)

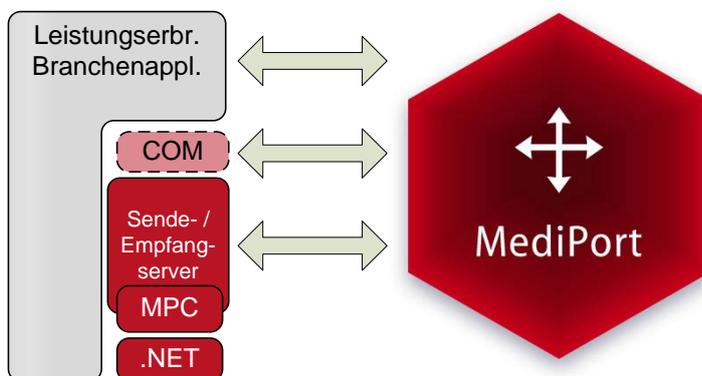


Der in der Grafik dargestellte Prozess der elektronischen Abrechnung zwischen Leistungserbringern und Kostenträgern gilt sowohl für elektronische Rechnungen, sowie auch für Rechnungsantworten in die umgekehrte Richtung.

## 1.2 Allgemein

Das Anbindungskonzept von Partnern ( Leistungserbringer / Kostenträger ) an MediPort sieht zur Zeit drei Möglichkeiten vor. Diese sind:

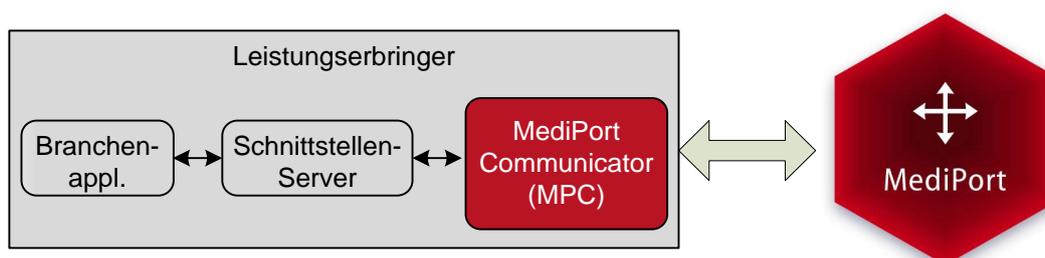
- COM-Komponenten (DLL's) für die Integration in die Branchenapplikation von Leistungserbringern(LE's)
- MediPort Gateway, Sende- / Empfangsserver designed für Kostenträger (KT's)
- Direktanschluss an den MediPort Server



Es hat sich gezeigt, dass die Lösung COM-Komponenten nicht für alle möglichen Leistungserbringer die optimale Lösung ist. Insbesondere im Spitalbereich kam vermehrt der Wunsch nach einem eigenständigen Sende- / Empfangsserver auf. Aber auch bei Leistungserbringern welche nicht Windows als Betriebssystem einsetzen, können die COM-Komponenten nicht verwendet werden und es braucht eine andere Lösung.

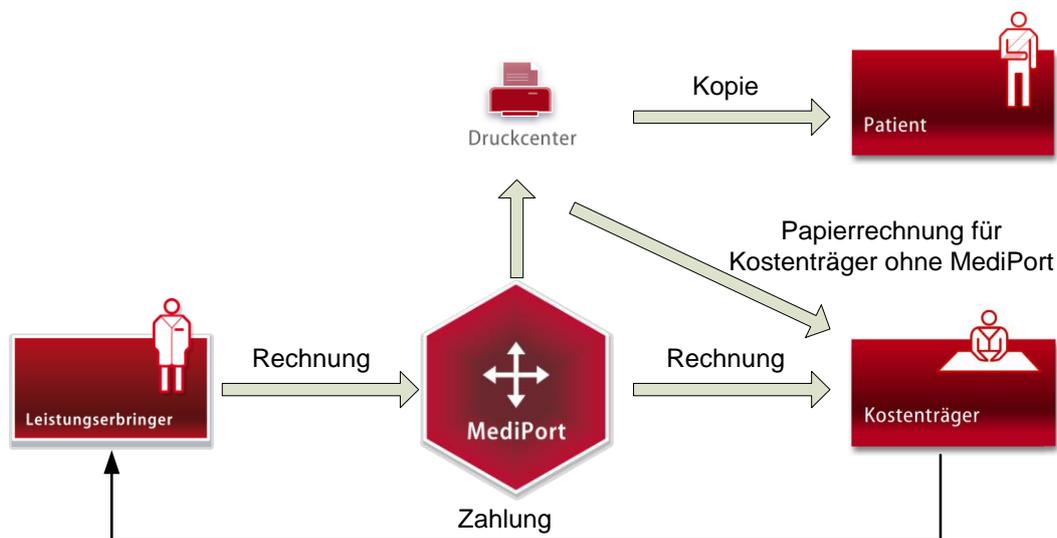
Die Lösung ist ein Sende- / Empfangsserver für Leistungserbringer mit dem Namen MediPort Communicator.

## 1.3 Positionierung



## 1.4 Workflow TP und TG

### 1.4.1 Workflow „Tiers Payant“ (TP)



Der Workflow Tiers Payant unterstützt die Zustellung von Dokumenten und Mahnungen im Tiers Payant Verfahren. Das Verfahren unterstützt den direkten Transport von Dokumenten zwischen Leistungserbringer und Versicherer. In Abhängigkeit vom Empfänger können folgende zwei Fälle unterschieden werden

#### **Empfänger ist Teilnehmer am MediPort System**

Das Dokument wird sofort in das Ausgangsfach des Empfängers gestellt und kann beim nächsten Abholvorgang vom Empfänger abgeholt werden. Die Zustellzeit ist minimal. Der Vorgang kann nicht storniert werden.

#### **Empfänger ist nicht Teilnehmer am MediPort System**

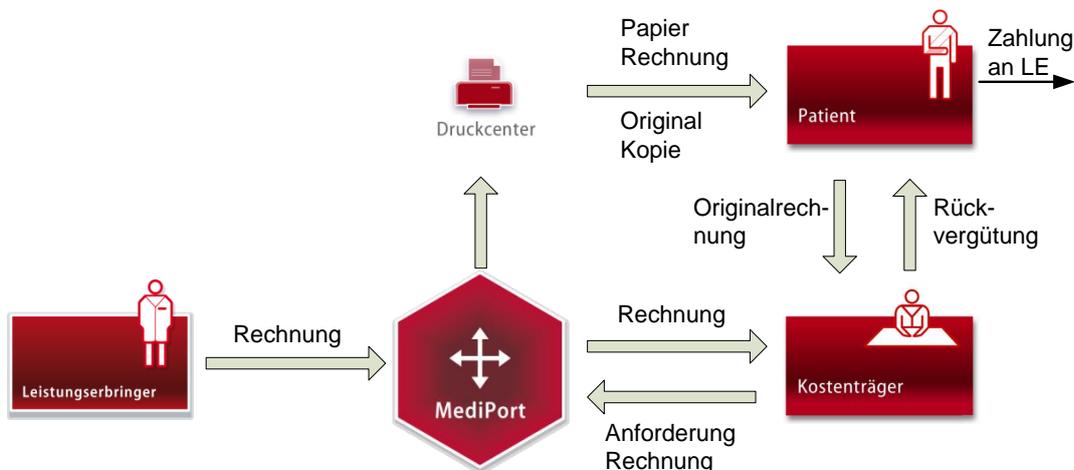
Das Dokument kann nicht elektronisch zugestellt werden. MediPort übergibt das Dokument dem Druckprozess und das Dokument wird dem Empfänger (i.d.R. einem Versicherer) auf dem Postweg zugestellt.

#### **Patient soll eine Orientierungskopie erhalten**

Der Absender kann mit jedem Versand eines TP-Dokuments wählen, ob eine Orientierungskopie an den Patienten gesandt werden soll oder nicht.

Ein Empfänger kann bei der Anmeldung oder zu einem späteren Zeitpunkt entscheiden, ob generell und automatisch von jedem TP-Dokument eine Orientierungskopie an den Patienten gedruckt und versandt werden soll oder nicht.

## 1.4.2 Workflow „Tiers Garant“ (TG)



Der Workflow Tiers Garant unterstützt die Zustellung von Dokumenten und Mahnungen im Tiers Garant Verfahren. Das Verfahren unterstützt die Zustellung von Dokumenten mit Hilfe einer Druckdienstleistung. Beim Workflow „Tiers Garant“ wird zwischen einer manuellen und einer automatischen Variante unterschieden.

Bei der manuellen Variante erfolgt in der Regel ein Ausdruck und Versand des Dokuments an die Adresse des Patienten, die vollständig im XML-Dokument vorhanden sein muss. Reicht der Patient das Dokument seinem Versicherer ein, so kann dieser den Datensatz mit dem aufgedruckten Barcode bei MediPort abholen. Reicht der Patient das Dokument nicht innerhalb 90 Tagen ein, so wird der Datensatz im MediPort Server unwiederbringlich gelöscht. Verlangt der Absender ausdrücklich keinen Druck und Versand, so wird angenommen, dass er das Dokument selbst druckt und die MediPort-Dokumentenummer selbst auf das Dokument anbringt, sodass der Datensatz von einem Versicherer abgerufen werden kann.

Ein TG-Dokument ist für den Empfänger „unsichtbar“. Erst unter Angabe der Empfängeradresse und der Dokumenten-Identifikation (Barcode) wird das Dokument für den Empfänger sichtbar. D.h. er muss im Besitze der Papierform des Dokuments sein.

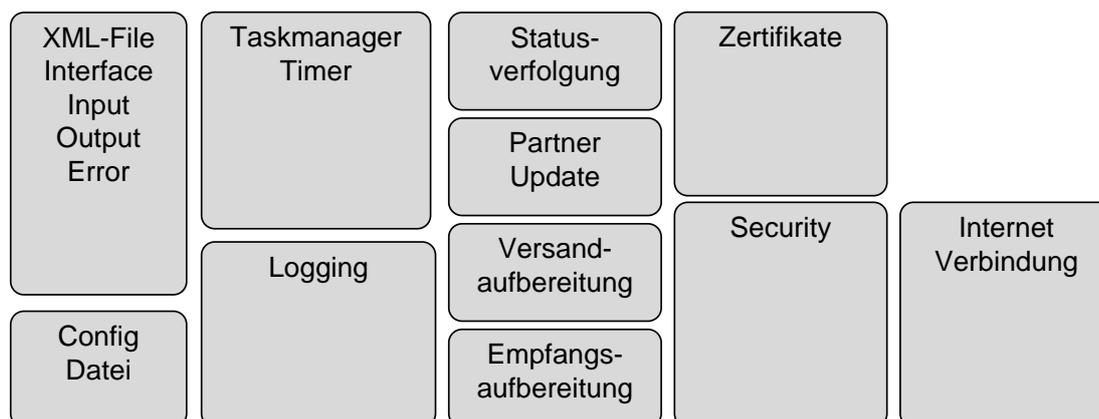
## 2. Grundlagen

### 2.1 Ziele

- Einfacher Anschluss an MediPort für Leistungserbringer
- Entkopplung Empfang/Versand <-> Verarbeitung
- Kapselung der Sicherheitsfeatures
- Anpassung an Netzwerkarchitektur
- Abhandlung aller Zustellarten
- Universelle Verwendung

### 2.2 Architektur

Die Architektur des MediPort Communicators sieht wie folgt aus:



Das Produkt ist nur für den Betrieb in gesicherten Netzen geeignet. (Hinter einer Firewall).

### 2.3 Systemanforderungen

#### 2.3.1 Unterstützte Betriebssysteme

Beim MediPort Communicator handelt es sich um eine „pure Java“ - Anwendung. Die Installation kann auf verschiedenen Betriebssystemen erfolgen:

- Microsoft Windows (Windows 7, 8, 10)
- Microsoft Windows Server 2008 R2 und 2012 R2 und 2016
- Mac OS X von 10.7.4 bis 10.12
- Unix-Derivate
  - Linux
  - Solaris x86 und SPARC ab OS 5.10 auf Anfrage

Andere auf Anfrage.

## 2.3.2 Hardwareanforderungen

Die Installation benötigt:

- 100 MByte Speicherplatz auf der Harddisk für Programme (inkl. Virtual Machine für Java)
- 100 MByte Speicherplatz für die Zwischenlagerung von Dateien (dies ist ein Richtwert und ist abhängig davon, welche Datenmenge gelagert werden soll. (Faustregel: 1 Rechnung in XML nach Tarmed: 6-10kByte; bei 100MByte können also cirka 10'000 Rechnungen zwischengelagert werden)
- Minimal 256 MByte RAM (Empfehlung: 512 MByte)
- Java Runtime Environment 1.8 (im Lieferumfang enthalten)

Für den Betrieb des MediPort Communicators ist ein Internet-Anschluss notwendig. Die notwendige Bandbreite richtet sich nach der Anzahl Dokumente die übertragen werden sollen. Mit 256kBit/s und einer Dokumentgröße von 10kByte könnten in der Stunde rund (256\*3600/150kBit) 6'000 Normdokumente durchgeschleust werden. Reserviert man 64kBit/s für die Anwendung MediPort so können damit cirka 1500 Normdokumente pro Stunde transferiert werden. In diesen Berechnungen ist die Zeit für die Authentifizierung von Server und Client sowie der Einfluss der Serverauslastung nicht berücksichtigt. Bei jedem Verbindungsaufbau werden die SSL-Parameter neu verhandelt, was eine gewisse Zeit in Anspruch nimmt.

Bitte beachten:

Die Bandbreite des eigenen Anschlusses ist nicht ausschließlich der limitierende Faktor im Internetbetrieb. Die Antwortzeiten des Domain Name Servers, des dienst anbietenden Servers sowie die Paketvermittlung sind ebenfalls zu berücksichtigen. Diese Zeiten sind ständigen Änderungen unterworfen und nur schwer zu berücksichtigen.

MediPort ist von der Architektur und den eingesetzten Produkten gut skalierbar und wächst mit den Anforderungen.

## 2.3.3 Anforderungen an Software

Der MediPort Communicator verwendet eine Java Virtual Machine 1.8, die im Lieferumfang enthalten ist.

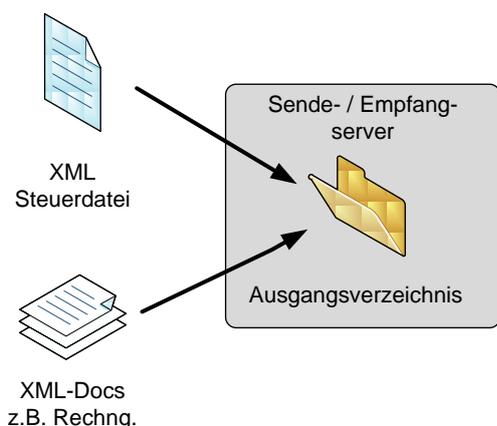
## 3. Funktionen

### 3.1 Daten senden

#### 3.1.1 Ablauf

- Die in einem Ausgangsverzeichnis befindlichen Dateien werden via MediPort an den Empfänger gesandt.
- Die Empfänger-EAN muss im Header der XML-Datei sein. Falls die Empfänger-EAN Nummer nicht im Partnerverzeichnis vorhanden ist (Empfänger ist nicht elektronisch an MediPort angeschlossen) wird automatisch der Druck des Dokumentes veranlasst.
- Es können XML-Dateien gemäss [www.forum-datenaustausch.ch](http://www.forum-datenaustausch.ch) versendet werden.
- Es können sowohl Tiers Payant sowie Tiers Garant Rechnungen verschickt werden.
- Vor dem Senden werden die XMLs schema-validiert, der Intermediär wird geprüft und ein wiederholtes Senden derselben XMLs wird unterbunden.
- Erfolgreich gesendete Dateien werden gelöscht.
- Fehlerhafte Dokumente werden in ein Fehlerverzeichnis geschrieben und aus dem Senderverzeichnis gelöscht.
- Versendete / empfangene Dateien werden in einer Log-Datei festgehalten.

Der MediPort Communicator benötigt zur Übermittlung von Dokumenten Zusatzinformationen, welche nicht aus der XML Datei ersichtlich sind. Diese sollen mit Hilfe einer Sende-Steuerdatei, die ins Ausgangsverzeichnis kopiert wird, dem MediPort Communicator übergeben werden. Es ist auch möglich, zuerst mehrere Sende-Steuerdateien ins Ausgangsverzeichnis zu stellen, welche dann nacheinander abgearbeitet werden.



Der Sendevorgang kann wie folgt ausgelöst werden:

- Der Communicator läuft als Applikation (alle Betriebssysteme), Service (Windows) oder Daemon (Unix/Linux) und der eingebaute Scheduler löst den Sendevorgang automatisch aus. Der Scheduler kann als einfacher Intervallscheduler oder als CronScheduler konfiguriert werden.
- Der Communicator läuft als RMI Server (alle Betriebssysteme) und der Sendevorgang wird von außerhalb getriggert.

## Hinweis:

Wichtig beim Kopieren in ein Sendeverzeichnis ist, dass eine Datei mit der Erweiterung **.tmp** erstellt und anschliessend umbenannt wird. Das Bereitstellen einer Datei muss also in 2 Schritten erfolgen. So ist sichergestellt, dass der Transport-Client die Datei nicht zu versenden beginnt, bevor sie fertig geschrieben (kopiert) ist.

1. **copy** <source-dir>/<filename> **to** <send-dir><filename.tmp>
2. **rename** <send-dir><filename.tmp> **to** <send-dir><filename>

Beim Verarbeiten einer XML Datei wird diese zunächst anhand des im schemaLocation-Attribut angegebenen Schemas validiert. Die unterstützten Schemas sind im <INSTALLDIR>/xsd/ Verzeichnis vorhanden. Pro XML werden nur die ersten 5 gefundenen Validierungsfehler ins Log (Kapitel 3.11) geschrieben. Danach wird geprüft, ob innerhalb der letzten Stunde bereits ein identisches Dokument versendet wurde. Falls in diesem Zeitraum mehr als 5000 Dokumente gesendet wurden, werden jedoch nur die neuesten 5000 XMLs mit dem aktuellen Dokument verglichen. Wenn ein XML mehr als 200 KB gross ist, wird für diese Prüfung auch nicht mehr das gesamte XML betrachtet, sondern lediglich die in der Sender-Steuerdatei angegebene SenderDoc ID verwendet. Schliesslich wird bei allen Dokumenten geprüft, ob als Intermediär die MediData-EAN 7601001304307 korrekt gesetzt ist. Per Konfiguration kann das automatische Korrigieren des Intermediärs eingeschaltet werden. In einem Fehlerfall wird das XML nicht gesendet, sondern in ein Fehlerverzeichnis verschoben (manche der in diesem Abschnitt genannten Werte sind per Konfiguration veränderbar und die Prüfungen können notfalls ausgeschaltet werden).

Besonderheiten der XML4.3/4.4 Dokumente:

- leere generalContainer\_430.xsd Dokumente können nicht gesendet werden
- wenn in einem generalContainer\_430.xsd Dokument eine Rechnung (generalInvoice[Request|Response]\_[430|440].xsd) vorhanden ist, wird der Empfänger hieraus ermittelt (unabhängig vom Empfänger in einem allfälligen MCD), aber wenn nur ein MCD vorhanden ist, gilt dessen Empfänger

verschlüsselte und/oder signierte Dokumente werden nicht speziell unterstützt

## 3.1.2 Sende-Steuerdatei

Bei der Sende-Steuerdatei handelt es sich um eine XML-Datei welche Informationen für 1 – n zu versendende Dateien enthalten kann. Pro zu versendender Datei gibt es einen Eintrag (Record) in der Datei mit folgenden Informationen:

- SenderDoc ID
- Filename der Datei zu welcher die Infos gehören
- DocAttr
- DocPrinted
- DistType
- Subject
- DocSize
- PrintLanguage

Zusätzliche optionale Informationen:

- TrustCenterEAN
- IsPaperInvoice
- SchemaID

Die Sende-Steuerdatei wird nach erfolgreichem Verarbeiten gelöscht, falls alle darin aufgeführten Dateien erfolgreich versandt wurden. Falls xmls ins Error-Verzeichnis verschoben wurden, wird die zugehörige Sende-Steuerdatei aber auch ins Error-Verzeichnis verschoben (wobei sie ggf. noch mit einem Suffix *.n.xml* (*n=integer Zahl*) umbenannt wird, um keine alte Sende-Steuerdatei zu überschreiben).

Der Name der Sende-Steuerdatei wird in der Konfigurationsdatei definiert. Default Name:

### **SendControl10.xml<sup>1</sup>**

#### **Beispiel für eine SendControl10.xml Datei mit 2 Dokumenten**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<DocumentsToSend xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="./data/send/SendControl10.xsd">
  <Document
    SenderDocId="Ihre einmalige Rechnungsnummer"
    FileName="abcdef.xml"
    DocAttr="Tiers_Payant"
    DocPrinted="true"
    DistType="0"
    Subject="Betrifftvermerk"
    DocSize="123"
    PrintLanguage="D"/>
  <Document
```

---

<sup>1</sup> Es besteht die Möglichkeit, die Records der einzelnen zu versendenden Dateien in ein File mit der Endung „\*.tmp“ zu schreiben und sobald der Batch-Prozess der Erstellung von Files abgeschlossen ist, kann der Dateiname geändert werden in SendControl\_10.xml und somit wird der Versand lanciert.

```
SenderDocId="Ref.723478a/34"  
FileName="xmit_23efdf.xml"  
DocAttr="Tiers_Payant"  
DocPrinted="true"  
DistType="1"  
Subject="Rechnung CSS"  
DocSize="4564"  
PrintLanguage="F"  
TrustCenterEAN="7601001370111"  
IsPaperInvoice="false"  
SchemaID="26020"/>  
</DocumentsToSend>
```

## XML – Definition (Schema)

Das File SendControl10.xsd welches die Vorschriften enthält, muss immer im Send Verzeichnis sein und darf dort nicht gelöscht werden!

Diese Vorschriften sind im Schema spezifiziert

### SenderDocId

Beliebige eindeutige Bezeichnung des Senders für das entsprechende Dokument.

### FileName

Name der Datei, welche zu versenden ist. Der Dateiname darf bis zu 32 Zeichen (Sonderzeichenregelung gemäss Betriebssystemen beachten) enthalten und muss mit der Zeichenfolge „.xml“ abgeschlossen sein.

### DocAttr

Definiert den Workflow auf MediPort und darf genau einen der folgenden Werte enthalten:

- Tiers\_Payant
- Tiers\_Garant\_Manuell
- Tiers\_Garant\_Direct (Nur für TG DocumentResponse, PullStandard)
- direct

### DocPrinted

Bedeutung je nach Workflow unterschiedlich:

- Bezeichnet im Tiers Garant, ob die Rechnung inklusive Rückforderungsbeleg gedruckt werden soll (Ausnahmen vorbehalten)
- Bezeichnet im Tiers Payant, ob die Patientenkopie gedruckt werden soll
- Bei direct irrelevant

Das Attribut muss einen der folgenden Werte erhalten:

„true“ = kein Druck

„false“ = druckt Rechnung/Rückforderungsbeleg/Patientenkopie gemäss Workflow

## DistType

Gibt die Versandart des gedruckten Dokuments an.

0 = B-Post (Default)

1 = A-Post

## Subject

Betreffvermerk ohne Umlaute

## DocSize

Zahl >= 0 gibt die Grösse der Datei in Bytes an

## PrintLanguage

Gibt an in welcher Sprache das Dokument gedruckt wird, falls es zum Druck kommt. Mögliche Werte „D“, „F“, „I“

## TrustCenterEAN

EAN Nummer des Trustcenters. Wenn diese gültig ist, wird eine Kopie des Dokumentes zusätzlich diesem Trustcenter zugestellt – Fehler -104 bei falscher EAN

Hinweis: Der Kunde muss zwingend einen Vertrag mit dem Trustcenter abschliessen, um Dokumente übermitteln zu können.

## IsPaperInvoice

Signalisiert, ob es sich um eine gescannte Papierrechnung handelt. Mögliche Werte: „true“, „false“

## SchemaID

Auf MediPort verwendete SchemaID, die zusätzlich zum xmit-Schema den Dokumententyp genauer spezifiziert. Dieser Parameter wird nur für Dokumente des Typs xmitdocuments\_410.xsd verwendet, bei anderen wird er ignoriert

## Die SendControl XML-Schema Datei (SendControl10.xsd)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema                                xmlns="http://www.medidata.ch/mpc/XSD"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.medidata.ch/mpc/XSD" elementFormDefault="qualified">
  <xs:complexType name="DocumentType">
    <xs:attribute name="SenderDocId" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string"/>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:schema>
```

```

</xs:attribute>
<xs:attribute name="FileName" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value=".{1,256}[.][x][m][l]" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="DocAttr" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="Tiers_Payant" />
      <xs:enumeration value="Tiers_Garant_Manuell" />
      <xs:enumeration value="Tiers_Garant_Direct" />
      <xs:enumeration value="direct" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="DocPrinted" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:boolean">
      <xs:pattern value="true|false" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="DistType" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0" />
      <xs:maxInclusive value="18" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="Subject" type="xs:string" use="optional"
default="XmlDocument" />
<xs:attribute name="DocSize" type="xs:long" use="required" />
<xs:attribute name="PrintLanguage" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="D" />
      <xs:enumeration value="F" />
      <xs:enumeration value="I" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="TrustCenterEAN" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="\d{13}" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="IsPaperInvoice" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:boolean">
      <xs:pattern value="true|false" />
    </xs:restriction>
  </xs:simpleType>

```

```
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="SchemaID" type="xs:string" use="optional"/>
</xs:complexType>
<xs:element name="DocumentsToSend">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Document" type="DocumentType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```

## 3.2 Daten empfangen

### 3.2.1 Allgemein

- Auf MediPort bereitliegende Dateien werden abgerufen und in ein Eingangsverzeichnis gespeichert.
- Die Dateien erhalten einen einmaligen, technischen Dateinamen.
- Es wird geprüft, ob als Intermediär die MediData-EAN 7601001304307 angegeben ist. Falls nicht, wird der Intermediär im Xml entsprechend abgeändert und ein WARN-Log-Eintrag erstellt.
- Empfangene Dateien werden in einer Log-Datei festgehalten.
- Empfangene Dateien können mit konfigurierbaren xslt Dateien automatisch transformiert werden.

Der Empfangsvorgang kann wie folgt ausgelöst werden:

- Der Communicator läuft als Applikation (alle Betriebssysteme), Service (Windows) oder Daemon(Unix/Linux) und der eingebaute Scheduler löst den Empfangsvorgang automatisch aus. Der Scheduler kann als einfacher Intervallscheduler oder als CronScheduler konfiguriert werden.
- Der Communicator läuft als RMI Server (alle Betriebssysteme) und der Empfangsvorgang wird von außerhalb getriggert.

Besonderheiten der XML4.3/4.4 und eKARUS Dokumente:

- Beim Empfang wird das Dokument mit genau einem transport / via Element mit der MediPort EAN 7601001304307 geschrieben. Ggf. wird das empfangene Dokument entsprechend angepasst. Bei generalContainer\_430.xsd Dokumenten gilt dies für Rechnung und MCD.

### 3.2.2 Schemaprüfung und automatischer Versand einer Antwort

Beim Empfang kann ein Dokument schemageprüft werden und wenn hierbei ein Schemafehler festgestellt wird, kann automatisch ein Antwortdokument generiert und an den Sender zurückgesandt werden und das empfangene Dokument wird in diesem Fall nicht in irgendeinem Empfangsverzeichnis gespeichert.

Für Tiers Garant Rechnungen werden keine Antworten generiert und versandt, die Schemaprüfung kann aber trotzdem ausgeführt werden.

Die Schemaprüfung ist standardmässig ausgeschaltet. Die Konfiguration ist in Kap. 5.2.4 beschrieben.

Für die Antwortgenerierung müssen einige Properties konfiguriert sein. Wenn eine Antwort generiert und versandt werden soll aber dies misslingt, weil etwa einige Properties falsch konfiguriert sind, so dass keine schemakonforme Antwort erstellt werden kann, wird das empfangene Dokument trotzdem verworfen, es wird einfach keine Antwort versandt.

Die Schemaprüfung beim Empfang kann – im Gegensatz zum Versand – nicht für alle Dokumenttypen, die empfangen werden können, ausgeführt werden, sondern höchstens für die nachfolgende Auswahl. Ferner wird nur für die nachfolgend mit einem Häkchen gekennzeichneten Schemas bei Bedarf eine automatische Antwort generiert und versendet:

Schemaname	Automat. Antwort?
LabInvoiceRequest_300	✓
MDInvoiceRequest_300	✓
HospitalInvoiceRequest_300	✓
InvoiceReminderRequest_300	✓
MDInvoiceRequest_400	✓
HospitalInvoiceRequest_400	✓
PharmacyInvoiceRequest_400	✓
InvoiceReminderRequest_400	✓
hospitalCreditRequest_410	✓
generalCreditRequest_410	✓
GeneralInvoiceRequest_410	✓
XmitDocuments_400	✗
XmitDocuments_410	✗
generalinvoicerequest_430	✓
hospitalmcdrequest_430	✓
statusrequest_430	✗
generalContainer_430	✓
generalCreditRequest_430	✓
generalNotification_430	✗
generalFormRequest_430	✓
generalFormResponse_430	✗
ekarus_x230	✓
ekarus_x240	✓
generalinvoicerequest_440	✓
hospitalmcdrequest_440	✓
generalNotification_440	✗

Der Dokumenttyp des im Fehlerfall generierten Antwortdokumentes entspricht immer dem Dokumenttyp des empfangenen Dokumentes, beispielsweise wird für ein Dokument des Typs *generalInvoiceRequest\_440.xsd* eine Antwort des Typs *generalInvoiceResponse\_440.xsd* erzeugt, ausser bei *generalContainer\_430.xsd*, wofür dessen Inhalt wie folgt massgebend ist:

Validierungsfehler	Antworttyp
nur in <b>generalinvoicerequest_4[3 4]0</b>	generalinvoiceresponse_4[3 4]0
nur in <b>hospitalmcdrequest_4[3 4]0</b>	hospitalmcdresponse_4[3 4]0
in <b>generalinvoicerequest_4[3 4]0</b> und <b>hospitalmcdrequest_4[3 4]0</b>	generalinvoiceresponse_4[3 4]0

Validierungsfehler	Antworttyp
in <b>ekarus_x230.xsd</b>	ekarus_x250.xsd
in <b>ekarus_x240.xsd</b>	hospitalmcdresponse_430
nur in <b>generalContainer_430</b>	keine Antwort

### 3.2.2.1 Konfigurationsparameter

Parameter	Beschreibung
mpcommunicator.receive.validation.enable	Schaltet die Validierung global für alle Schemen ein/aus
mpcommunicator.receive.validation.LabInvoiceRequest_300	Schaltet die Validierung für das Schema LabInvoiceRequest_300 ein/aus
mpcommunicator.receive.validation.MDInvoiceRequest_300	Schaltet die Validierung für das Schema MDInvoiceRequest_300 ein/aus
mpcommunicator.receive.validation.HospitalInvoiceRequest_300	Schaltet die Validierung für das Schema HospitalInvoiceRequest_300 ein/aus
mpcommunicator.receive.validation.InvoiceReminderRequest_300	Schaltet die Validierung für das Schema InvoiceReminderRequest_300 ein/aus
mpcommunicator.receive.validation.MDInvoiceRequest_400	Schaltet die Validierung für das Schema MDInvoiceRequest_400 ein/aus
mpcommunicator.receive.validation.HospitalInvoiceRequest_400	Schaltet die Validierung für das Schema HospitalInvoiceRequest_400 ein/aus
mpcommunicator.receive.validation.PharmacyInvoiceRequest_400	Schaltet die Validierung für das Schema PharmacyInvoiceRequest_400 ein/aus
mpcommunicator.receive.validation.InvoiceReminderRequest_400	Schaltet die Validierung für das Schema InvoiceReminderRequest_400 ein/aus
mpcommunicator.receive.validation.HospitalCreditRequest_410	Schaltet die Validierung für das Schema HospitalCreditRequest_410 ein/aus
mpcommunicator.receive.validation.GeneralCreditRequest_410	Schaltet die Validierung für das Schema GeneralCreditRequest_410 ein/aus
mpcommunicator.receive.validation.GeneralInvoiceRequest_410	Schaltet die Validierung für das Schema GeneralInvoiceRequest_410 ein/aus
mpcommunicator.receive.validation.XmitDocuments_400	Schaltet die Validierung für das Schema XmitDocuments_400 ein/aus
mpcommunicator.receive.validation.XmitDocuments_410	Schaltet die Validierung für das Schema XmitDocuments_410 ein/aus
mpcommunicator.receive.validation.GeneralInvoiceRequest_430	Schaltet die Validierung für das Schema GeneralInvoiceRequest_430 ein/aus
mpcommunicator.receive.validation.HospitalMcdRequest_430	Schaltet die Validierung für das Schema HospitalMcdRequest_430 ein/aus
mpcommunicator.receive.validation.StatusRequest_430	Schaltet die Validierung für das Schema StatusRequest_430 ein/aus
mpcommunicator.receive.validation.GeneralContainer_430	Schaltet die Validierung für das Schema GeneralContainer_430 ein/aus
mpcommunicator.receive.validation.GeneralCreditRequest_430	Schaltet die Validierung für das Schema GeneralCreditRequest_430 ein/aus
mpcommunicator.receive.validation.GeneralNotification_430	Schaltet die Validierung für das Schema GeneralNotification_430 ein/aus
mpcommunicator.receive.validation.GeneralFormRequest_430	Schaltet die Validierung für das Schema GeneralFormRequest_430 ein/aus
mpcommunicator.receive.validation.GeneralFormResponse_430	Schaltet die Validierung für das Schema GeneralFormResponse_430 ein/aus
mpcommunicator.receive.validation.Ekarus_x230	Schaltet die Validierung für das Schema Ekarus_x230 ein/aus
mpcommunicator.receive.validation.Ekarus_x240	Schaltet die Validierung für das Schema Ekarus_x240 ein/aus
mpcommunicator.receive.validation.GeneralInvoiceRequest_440	Schaltet die Validierung für das Schema GeneralInvoiceRequest_440 ein/aus
mpcommunicator.receive.validation.HospitalMcdRequest_440	Schaltet die Validierung für das Schema HospitalMcdRequest_440 ein/aus

mpcommunicator.receive.validation.GeneralNotification_440	Schaltet die Validierung für das Schema GeneralNotification_440 ein/aus
mpcommunicator.receive.validation.responsegenerator_switch	Schaltet die Response Generierung für alle Schemen ein/aus
mpcommunicator.receive.validation.responsegenerator.name	Definiert der Name des Response Generators, falls das Dokument nicht konform ist. (nur für TP Rechnungen relevant)
mpcommunicator.receive.validation.responsegenerator.version	Definiert die Version des Response Generators, falls das Dokument nicht konform ist. (nur für TP Rechnungen relevant)
mpcommunicator.receive.validation.contact.companyname	Kontaktdaten, falls eine Rechnung zurückgewiesen wird.
mpcommunicator.receive.validation.contact.department	Kontaktdaten, falls eine Rechnung zurückgewiesen wird.
mpcommunicator.receive.validation.contact.zip	Kontaktdaten, falls eine Rechnung zurückgewiesen wird.
mpcommunicator.receive.validation.contact.city	Kontaktdaten, falls eine Rechnung zurückgewiesen wird.
mpcommunicator.receive.validation.contact.email	Kontaktdaten, falls eine Rechnung zurückgewiesen wird.
mpcommunicator.receive.validation.contact.phone	Kontaktdaten, falls eine Rechnung zurückgewiesen wird.

### 3.2.3 Überschreiben von recipient und insurance

Beim Empfang können für einige Dokumenttypen die recipient-GLN sowie die insurance-Daten verändert werden, bevor das empfangene Dokument in ein Eingangsverzeichnis gespeichert wird. Dies kann für die folgenden Dokumenttypen konfiguriert werden:

Schemaname	recipient ean	insurance data
LabInvoiceRequest_300	✓	✓
MDInvoiceRequest_300	✓	✓
HospitalInvoiceRequest_300	✓	✓
InvoiceReminderRequest_300	✓	✓
MDInvoiceRequest_400	✓	✓
HospitalInvoiceRequest_400	✓	✓
PharmacyInvoiceRequest_400	✓	✓
InvoiceReminderRequest_400	✓	✓
hospitalCreditRequest_410	✓	✓
generalCreditRequest_410	✓	✓
GeneralInvoiceRequest_410	✓	✓
XmitDocuments_400	✗	✗
XmitDocuments_410	✗	✗
generalinvoicerequest_430	✓	✓
hospitalmcdrequest_430	✓	✓
statusrequest_430	✓	✗
generalContainer_430	✓	✓
generalCreditRequest_430	✓	✓
generalNotification_430	✓	✓

Schemaname	recipient ean	insurance data
generalFormRequest_430	✓	✓
generalFormResponse_430	✓	✗
ekarus_x230.xsd	✓	✓
ekarus_x240.xsd	✓	✓
generalinvoicerequest_440	✓	✓
hospitalmcdrequest_440	✓	✓
generalNotification_440	✓	✓

Anpassungen werden nur an schemakonformen Dokumenten vorgenommen, d.h. unabhängig von der Funktion "Schemaprüfung und automatischer Versand einer Antwort" werden Dokumente, die anzupassen sind, validiert. Nach der Anpassung ist das empfangene Dokument jedoch nicht zwingend schemakonform, etwa bei fehlerhaften Angaben im entsprechenden Konfigurationsfile.

Das Überschreiben von recipient und insurance ist standardmässig ausgeschaltet. Die Konfiguration für diese Funktion ist in Kap. 5.2.5 beschrieben.

Die Funktionen "Schemaprüfung und automatischer Versand einer Antwort" sowie "Überschreiben von recipient und insurance" können unabhängig voneinander eingesetzt werden. Das Überschreiben von recipient und insurance geschieht jedoch nach der Funktion "Schemaprüfung und automatischer Versand einer Antwort". Falls diese bereits zum Verwerfen des empfangenen Dokumentes führt, ist die Funktion des Überschreibens also hinfällig.

### 3.2.3.1 Konfigurationsparameter

Parameter	Beschreibung
mpcommunicator.receive.modification.override_recipient_ean.enable	Schaltet die Recipient-EAN-Korrektur global ein/aus
mpcommunicator.receive.modification.override_insurance_data.enable	Schaltet die Insurance-Addressierungskorrektur global ein/aus
mpcommunicator.receive.modification.recipient_ean	EAN Wert für die Recipient-Korrektur
mpcommunicator.receive.modification.insurance_ean	EAN Wert für die Insurance-Korrektur
mpcommunicator.receive.modification.insurance_company_name	Versicherungsname für die Insurance-Korrektur
mpcommunicator.receive.modification.insurance_street	Strasse für die Insurance-Korrektur
mpcommunicator.receive.modification.insurance_zip	Postleitzahl für die Insurance-Korrektur
mpcommunicator.receive.modification.insurance_city	Ortschaft für die Insurance-Korrektur
mpcommunicator.receive.modification.recipient.LabInvoiceRequest_300	Schaltet die Recipient-Korrektur für das Schema LabInvoiceRequest_300 ein/aus
mpcommunicator.receive.modification.recipient.MDInvoiceRequest_300	Schaltet die Recipient-Korrektur für das Schema MDInvoiceRequest_300 ein/aus

mpcommunicator.receive.modification.recipient.HospitalInvoiceRequest_300	Schaltet die Recipient-Korrektur für das Schema HospitalInvoiceRequest_300 ein/aus
mpcommunicator.receive.modification.recipient.InvoiceReminderRequest_300	Schaltet die Recipient-Korrektur für das Schema InvoiceReminderRequest_300 ein/aus
mpcommunicator.receive.modification.recipient.MDInvoiceRequest_400	Schaltet die Recipient-Korrektur für das Schema MDInvoiceRequest_400 ein/aus
mpcommunicator.receive.modification.recipient.HospitalInvoiceRequest_400	Schaltet die Recipient-Korrektur für das Schema HospitalInvoiceRequest_400 ein/aus
mpcommunicator.receive.modification.recipient.PharmacyInvoiceRequest_400	Schaltet die Recipient-Korrektur für das Schema PharmacyInvoiceRequest_400 ein/aus
mpcommunicator.receive.modification.recipient.InvoiceReminderRequest_400	Schaltet die Recipient-Korrektur für das Schema InvoiceReminderRequest_400 ein/aus
mpcommunicator.receive.modification.recipient.HospitalCreditRequest_410	Schaltet die Recipient-Korrektur für das Schema HospitalCreditRequest_410 ein/aus
mpcommunicator.receive.modification.recipient.GeneralCreditRequest_410	Schaltet die Recipient-Korrektur für das Schema GeneralCreditRequest_410 ein/aus
mpcommunicator.receive.modification.recipient.GeneralInvoiceRequest_410	Schaltet die Recipient-Korrektur für das Schema GeneralInvoiceRequest_410 ein/aus
mpcommunicator.receive.modification.recipient.GeneralInvoiceRequest_430	Schaltet die Recipient-Korrektur für das Schema GeneralInvoiceRequest_430 ein/aus
mpcommunicator.receive.modification.recipient.HospitalMcdRequest_430	Schaltet die Recipient-Korrektur für das Schema HospitalMcdRequest_430 ein/aus
mpcommunicator.receive.modification.recipient.StatusRequest_430	Schaltet die Recipient-Korrektur für das Schema StatusRequest_430 ein/aus
mpcommunicator.receive.modification.recipient.GeneralContainer_430	Schaltet die Recipient-Korrektur für das Schema GeneralContainer_430 ein/aus
mpcommunicator.receive.modification.recipient.GeneralCreditRequest_430	Schaltet die Recipient-Korrektur für das Schema GeneralCreditRequest_430 ein/aus
mpcommunicator.receive.modification.recipient.GeneralNotification_430	Schaltet die Recipient-Korrektur für das Schema GeneralNotification_430 ein/aus
mpcommunicator.receive.modification.recipient.GeneralFormRequest_430	Schaltet die Recipient-Korrektur für das Schema GeneralFormRequest_430 ein/aus
mpcommunicator.receive.modification.recipient.GeneralFormResponse_430	Schaltet die Recipient-Korrektur für das Schema GeneralFormResponse_430 ein/aus

mpcommunicator.receive.modification.recipient.Ekarus_x230	Schaltet die Recipient-Korrektur für das Schema Ekarus_x230 ein/aus
mpcommunicator.receive.modification.recipient.Ekarus_x240	Schaltet die Recipient-Korrektur für das Schema Ekarus_x240 ein/aus
mpcommunicator.receive.modification.recipient.GeneralInvoiceRequest_440	Schaltet die Recipient-Korrektur für das Schema GeneralInvoiceRequest_440 ein/aus
mpcommunicator.receive.modification.recipient.HospitalMcdRequest_440	Schaltet die Recipient-Korrektur für das Schema HospitalMcdRequest_440 ein/aus
mpcommunicator.receive.modification.recipient.GeneralNotification_440	Schaltet die Recipient-Korrektur für das Schema GeneralNotification_440 ein/aus
mpcommunicator.receive.modification.insurance.LabInvoiceRequest_300	Schaltet die Insurance-Korrektur für das Schema LabInvoiceRequest_300 ein/aus
mpcommunicator.receive.modification.insurance.MDInvoiceRequest_300	Schaltet die Insurance-Korrektur für das Schema MDInvoiceRequest_300 ein/aus
mpcommunicator.receive.modification.insurance.HospitalInvoiceRequest_300	Schaltet die Insurance-Korrektur für das Schema HospitalInvoiceRequest_300 ein/aus
mpcommunicator.receive.modification.insurance.InvoiceReminderRequest_300	Schaltet die Insurance-Korrektur für das Schema InvoiceReminderRequest_300 ein/aus
mpcommunicator.receive.modification.insurance.MDInvoiceRequest_400	Schaltet die Insurance-Korrektur für das Schema MDInvoiceRequest_400 ein/aus
mpcommunicator.receive.modification.insurance.HospitalInvoiceRequest_400	Schaltet die Insurance-Korrektur für das Schema HospitalInvoiceRequest_400 ein/aus
mpcommunicator.receive.modification.insurance.PharmacyInvoiceRequest_400	Schaltet die Insurance-Korrektur für das Schema PharmacyInvoiceRequest_400 ein/aus
mpcommunicator.receive.modification.insurance.InvoiceReminderRequest_400	Schaltet die Insurance-Korrektur für das Schema InvoiceReminderRequest_400 ein/aus
mpcommunicator.receive.modification.insurance.HospitalCreditRequest_410	Schaltet die Insurance-Korrektur für das Schema HospitalCreditRequest_410 ein/aus
mpcommunicator.receive.modification.insurance.GeneralCreditRequest_410	Schaltet die Insurance-Korrektur für das Schema GeneralCreditRequest_410 ein/aus
mpcommunicator.receive.modification.insurance.GeneralInvoiceRequest_410	Schaltet die Insurance-Korrektur für das Schema GeneralInvoiceRequest_410 ein/aus
mpcommunicator.receive.modification.insurance.GeneralInvoiceRequest_430	Schaltet die Insurance-Korrektur für das Schema GeneralInvoiceRequest_430 ein/aus

mpcommunicator.receive.modification.insurance.HospitalMcdRequest_430	Schaltet die Insurance-Korrektur für das Schema HospitalMcdRequest_430 ein/aus
mpcommunicator.receive.modification.insurance.StatusRequest_430	Schaltet die Insurance-Korrektur für das Schema StatusRequest_430 ein/aus
mpcommunicator.receive.modification.insurance.GeneralContainer_430	Schaltet die Insurance-Korrektur für das Schema GeneralContainer_430 ein/aus
mpcommunicator.receive.modification.insurance.GeneralCreditRequest_430	Schaltet die Insurance-Korrektur für das Schema GeneralCreditRequest_430 ein/aus
mpcommunicator.receive.modification.insurance.GeneralNotification_430	Schaltet die Insurance-Korrektur für das Schema GeneralNotification_430 ein/aus
mpcommunicator.receive.modification.insurance.GeneralFormRequest_430	Schaltet die Insurance-Korrektur für das Schema GeneralFormRequest_430 ein/aus
mpcommunicator.receive.modification.insurance.GeneralFormResponse_430	Schaltet die Insurance-Korrektur für das Schema GeneralFormResponse_430 ein/aus
mpcommunicator.receive.modification.insurance.Ekarus_x230	Schaltet die Insurance-Korrektur für das Schema Ekarus_x230 ein/aus
mpcommunicator.receive.modification.insurance.Ekarus_x240	Schaltet die Insurance-Korrektur für das Schema Ekarus_x240 ein/aus
mpcommunicator.receive.modification.insurance.GeneralInvoiceRequest_440	Schaltet die Insurance-Korrektur für das Schema GeneralInvoiceRequest_440 ein/aus
mpcommunicator.receive.modification.insurance.HospitalMcdRequest_440	Schaltet die Insurance-Korrektur für das Schema HospitalMcdRequest_440 ein/aus
mpcommunicator.receive.modification.insurance.GeneralNotification_440	Schaltet die Insurance-Korrektur für das Schema GeneralNotification_440 ein/aus

### 3.2.4 MultiUser

Ab MPC V1.5.0 gibt es eine MultiUser Funktionalität die nicht nur auf den Empfang beschränkt ist. Näheres dazu im Kapitel 3.8.

### 3.2.5 XSLT Transformation

Pro Mandant und pro Typ (xsd Schema) der empfangenen Datei kann eine xslt Datei konfiguriert werden, mit der ein empfangenes Dokument transformiert wird. Die transformierte Datei erhält den Namen des empfangenen Dokumentes, welches nicht gespeichert wird. Die Zuordnung von Schema und xslt Datei, welche für ein Schema anzuwenden ist, soll über eine jeweils identische Integer-Zahl  $m$  erfolgen, so dass pro Mandant und Rechnungstyp je 2 Properties wie folgt zu konfigurieren sind (im Übrigen verhält sich die Konfiguration entsprechend den allgemeinen Parametern pro Mandant gemäss Kap. 5.2.6):

mpcommunicator.config (optional)		
Key	Beispiel Value	Default
CLIENT.n.XSLT.m.DOCTYPE	generallInvoiceResponse_430.xsd	
CLIENT.n.XSLT.m.XSLT	data/xslt/generallInvoiceResponse_430.xslt	

## Hinweis:

- Konfigurierte Schemas werden in Kleinschreibung abgespeichert, so dass Gross/Kleinschreibung des Schemas im empfangenen Dokument unerheblich ist
- Die `CLIENT.n.XSLT.m.DOCTYPE` Property soll nur der Dateiname der Schemabezeichnung (gemäss Beispiel ohne schemaLocation mit http:// etc.) enthalten, die `CLIENT.n.XSLT.m.XSLT` Property soll auch den relativen oder absoluten Pfad inklusive des xslt-Dateinamens enthalten

Mit `mpcommunicator.receive.xslt.deleteorig=false` kann ggf. das automatische Löschen des empfangenen Dokumentes unterbunden werden, dieses hat dann bis auf die letzten Stellen vor dem Suffix (Zeitstempel) denselben Dateinamen wie das transformierte Dokument.

### 3.2.6 eKARUS zu XML 4.3 Konvertierung

Die Verwendung dieser Funktion muss von MediData freigegeben werden. Es wird bisher nur zu XML 4.3, nicht XML 4.4 konvertiert.

Folgende Konvertierungen sind beim Empfang möglich:

- `ekarus_x230.xsd` → `generallInvoiceRequest_430.xsd`
- `ekarus_x240.xsd` → `hospitalMCDRequest_430.xsd`
- `generalContainer_430.xsd` mit `ekarus_x230.xsd` und/oder `ekarus_x240.xsd` → `generalContainer_430.xsd` mit `generallInvoiceRequest_430.xsd` und/oder `hospitalMCDRequest_430.xsd`

Folgende Parameter sind zu konfigurieren. Analog zu den Empfangsverzeichnissen pro Mandant (Kap. 5.2.8) sind für die eKARUS-Originaldokumente zusätzliche Verzeichnisse pro Mandant anzugeben, die sich von den anderen Empfangsverzeichnissen unterscheiden, welche die konvertierten Dokumente enthalten. Ferner muss die Funktion eingeschaltet werden:

mpcommunicator.config (optional)		
Key	Bedeutung	Default
CLIENT.n.RECEIVEORIG_DIR	Empfangsverzeichnis Originale produktive Dokumente	
CLIENT.n.RECEIVEORIGTEST_DIR	data/xslt/generallInvoiceResponse_430.xslt	
mpcommunicator.receive.ekarusconversion.enable	eKARUS Konvertierung ein/aus mit true oder false	false

## 3.3 Partnerverzeichnis

### 3.3.1 Allgemein

- Die Kommunikationspartnerangaben werden von MediPort geladen und in einer Textdatei gespeichert.
- Der Inhalt dieser Datei muss von der Branchenapplikation ausgewertet werden damit fehlerhafte XML-Dokumente (nicht bekannte Empfänger) vermieden werden.
- Der Zeitintervall in dem das Partnerverzeichnis upgedated wird, kann in der Konfigurationsdatei eingestellt werden.

Das Nachführen des Partnerverzeichnis kann wie folgt ausgelöst werden:

- Der Communicator läuft als Applikation (alle Betriebssysteme), Service (Windows) oder Daemon (Unix/Linux) und der eingebaute Scheduler löst den Empfangsvorgang automatisch aus. Der Scheduler kann als einfacher Intervallscheduler oder als CronScheduler konfiguriert werden.

Der Communicator läuft als RMI Server (alle Betriebssysteme) und das Nachführen des Partnerverzeichnis wird von außerhalb getriggert.

#### Hinweis:

Ab Version 1.9.0 des MPC wird das Partnerverzeichnis bei einem Wechsel des verwendeten MediPort Servers (Property mpcommunicator.mediport.ip) automatisch korrekt neu gebildet (vorher musste hierbei die Datei config/db/mpcommunicator.script mit der default-Datei im config/default/db/ Verzeichnis ersetzt werden).

## 3.3.2 Aufbau des Partnerverzeichnisses

Das Partnerverzeichnis ist wie folgt aufgebaut:

```
<Header>
<Entry1>
<Entry1>
.....
<Entryn>
```

### Beispiel eines erzeugten Partnerverzeichnisses

```
EAN;ORGANISATION;DEPARTMENT;
7601000173003;Concordia;Vertrauensaerztlicher Dienst;
7601003000078;Suva;Test;
```

## 3.4 Statusverfolgung

### 3.4.1 Funktionsbeschreibung

Der Scheduler des MPC startet den Task „GetDocumentList“ periodisch aufgrund der Konfiguration. Der MPC merkt sich die Zeit der Aktualisierung. Die Serverantwort wird in Form einer XML-Datei mit einem sprechenden Dateinamen in ein bestimmtes Verzeichnis abgespeichert. Die Branchenapplikation (BA) verarbeitet in regelmäßigen Abständen die neu erhaltenen Dateien und löscht diese nach dem Verarbeiten. Die Verarbeitung der BA beinhaltet im Wesentlichen die Nachführung des Status.

### 3.4.2 Rückgabedatei

#### Ablageort und Namensgebung

Die Serverantwort wird in Dateiform in das Verzeichnis <INSTALL\_DIR>/data/docstatus gelegt. Der Dateiname wird nach folgendem Namensschema angelegt:

#### Namensschema

ML\_YYMMDD\_yymmddhhmmss\_NNN.XML wobei:

- M Dateikennzeichnung
- YYMMDD Abfragestartdatum
- yymmddhhmmss Abfrageenddatum(mit Zeitangabe)
- NNN Fortlaufende Nummerierung bei mehr als 1000 Dokumentenstatus
- XML Dateiendung

## 3.4.3 XML Schema

Der Inhalt wird gemäss folgendem XML-Schema in Dateiform abgelegt.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="GetDocListResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Document" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Document">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="AugmentedDocs" type="AugmentedDocInfo" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="DateReceived" type="xs:string" use="required"/>
      <xs:attribute name="MDID" type="xs:unsignedLong" use="required"/>
      <xs:attribute name="DocStatus" type="xs:byte" use="required"/>
      <xs:attribute name="StatusDate" type="xs:string" use="required"/>
      <xs:attribute name="SenderDocumentID" type="xs:string" use="optional"/>
      <xs:attribute name="Subject" type="xs:string" use="optional"/>
      <xs:attribute name="DocSize" type="xs:unsignedLong" use="optional"/>
      <xs:attribute name="Role" type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="AugmentedDocInfo">
    <xs:attribute name="Receivertyp" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="TrustCenter"/>
          <xs:enumeration value="Patient"/>
          <xs:enumeration value="Partner"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Receiveraddress" type="xs:string" use="optional"/>
    <xs:attribute name="Status" type="xs:byte" use="required"/>
    <xs:attribute name="MDID" type="xs:unsignedLong" use="optional"/>
  </xs:complexType>
</xs:schema>
```

### Attributliste von „Document“

Parameter	Bedeutung	Hinweis	Werteliste
DateReceived	Versanddatum	„DD.MM.YYYY HH:MM“	–

MDID	MediPort interne Identifikation	Zehnstellige Nummer	Zehnstellige Nummer mit 1xxxxxxx
DocStatus	Status des Documents		Gemäss Statustabelle
StatusDate	Zeitstempel der Statusänderung	„DD.MM.YYYY – HH:MM“	
SenderDocumentID	ID des Sender (meist Rechnungsnummer)	Muss pro Sender eindeutig sein	
Subject	Bezeichnung des Senders	Freier Text	
DocSize	Dokumentgrösse	In Bytes	
Role	Test oder Produktiv		test oder production

## Attributliste von „AugmentedDocInfo“

Parameter	Bedeutung	Hinweis	Werteliste
Receivertyp	Gibt an, an welchen Partnertyp die Kopie gesandt wurde	Partner ist inaktiv und für künftige Erweiterungen	Trustcenter / Patient / Partner
Receiveraddress	EAN Nummer des Empfängers		EAN der Trustcenter EAN MediPort Trustcenter
Status	Status des Documents		Gemäss Statustabelle
MDID	MediPort interne ID für die Kopie	Wird im Moment nicht zurückgeliefert	Zehnstellige Nummer

## Statustabelle

Codes für den Dokumentenstatus (Range 1 bis 99)		
Code	Bezeichnung	Beschreibung
1	In MediPort-Server-Inbox eingetroffen	Das Dokument wurde empfangen und von MediPort der Empfang bestätigt
2	In Verarbeitung	Das Dokument befindet sich in der ordentlichen Verarbeitung
3	TG fertig	Das Dokument wurde als Tiers Garant fertig prozessiert und ist

	prozessiert nicht abgeholt	bereit, um abgeholt zu werden.
4	Storniert	Das Dokument wurde vom Sender storniert.
5	Abgeholt fertig prozessiert	Das Dokument wurde fertig prozessiert (Vom Empfänger elektronisch abgeholt oder gedruckt)
98	Nicht zustellbar via Printoutput	Das Dokument konnte nicht gedruckt werden. In der Regel fehlt die Zustelladresse
99	Nicht zustellbar	Das Dokument konnte nicht wie gewünscht zugestellt werden

## 3.5 Senden ohne Sende-Steuerdateien

### 3.5.1 Funktionsbeschreibung

Ab Version 1.9.0 kann der MPC auch Senden, ohne dass Sende-Steuerdateien (Kapitel 3.1.2) bereitgestellt werden müssen. Stattdessen können die gewünschten Parameter spezifiziert werden, oder der MPC verwendet Default-Werte, danach erstellt der MPC die notwendigen Sende-Steuerdateien selber.

Statt wie bisher in das Sende-Verzeichnis eines Mandanten sollen die zu sendenden xmls in konfigurierte Verzeichnisse abgelegt werden. Für jedes solche Verzeichnis können zugehörige Parameter festgelegt werden, nach welchen der MPC dann automatisch entsprechende Sende-Steuerdateien generiert.

Das Erstellen von Sende-Steuerdateien kann auch direkt von einer etwaigen generischen Funktion, d.h. einem Plugin (Kap. 3.7), kontrolliert werden, welches Sende-Steuerdateien so flexibler statt nur fix vorkonfiguriert erstellen kann, etwa mit zusätzlichen nur zur Laufzeit erhältlichen Informationen zu Schemaid.

### 3.5.2 Konfiguration der Parameter

In der Konfigurationsdatei `mpcommunicator.config` kann pro Mandant eine beliebige Anzahl von Sende-Parameter-Sets konfiguriert werden, nach welchen der MPC dann automatisch entsprechende Sende-Steuerdateien erstellt.

#### Hinweis:

- Die Parameter eines Parameter-Sets müssen jeweils identische Werte `CLIENT.n.DIR.m` im Key enthalten, wobei  $n$  und  $m$  beliebige (positive) Integer-Werte sein dürfen und  $n$  den betreffenden `CLIENT` identifiziert (mittels `CLIENT.n.EAN` spezifiziert) und  $m$  das Parameter-Set kennzeichnet.

- Ein Parameter-Set muss mindestens den ersten Parameter `CLIENT.n.DIR.m` angeben. Mit diesem Parameter wird das Verzeichnis angegeben, in welchem die zu senden xmls bereitgestellt werden sollen. Dieser Pfad kann relativ (vgl. Beispiel unten) oder absolut (z.B. `CLIENT.50.DIR.100=D:/data/mpctests/testclient`) angegeben werden. Xmls, die unvollständig sind, weil das Dateisystem vielleicht gerade am Schreiben der Datei ist, werden einfach ignoriert. Da die Sende-Steuerdatei nur Xmls mit der Dateiendung ".xml" zulässt, werden Xmls mit den Endungen ".XML" oder ".Xml" auf ".xml" abgeändert. Andere Dateien werden ignoriert.
- **Wichtig:** Wenn für die Konfiguration der Sende-, Empfangs-, Error- und Archivverzeichnisse bereits das `smb://` Protokoll eingesetzt wird, muss auch der Parameter `CLIENT.n.DIR.m` als smb-Verzeichnis konfiguriert werden (vgl. Kap. 5.3.3).
- Wenn der fakultative Parameter `CLIENT.n.DIR.m.PLUGIN` vorhanden ist, wird diese Parameter-Set-Konfiguration von dieser MPC-Funktion ignoriert und nur von einer etwaig vorhandenen generischen Funktion mit dem angegebenen Namen verarbeitet.
- Die übrigen 7 Parameter sind fakultativ und spezifizieren die gleichnamigen Attribute der zu erstellenden Sende-Steuerdateien gemäss Kapitel 3.1.2, und ihre Werte sollen ebenso gesetzt werden.
- Wenn fakultative Parameter nicht konfiguriert werden, verwendet der MPC für `DOCPRINTED`, `DISTTYPE` und `PRINTLANGUAGE` die angegebenen Default-Werte. Für die übrigen Parameter verfährt der MPC wie folgt:
- `DOCATTR`: Bei allen Dokumenttypen in der nachfolgenden Tabelle, wo **DocAttr** direct angegeben ist, wird `DocAttr` automatisch auf `direct` gesetzt (falls vorhanden, wird eine davon abweichende Konfiguration in diesem Fall mit einem ERROR gemeldet). Bei allen übrigen Dokumenttypen hat ein konfigurierter Wert Vorrang, ansonsten wird das xml gelesen: wenn im xml ein Element `invoice:tiers_payant` vorkommt, wird `DocAttr` auf `Tiers_Payant` gesetzt, oder wenn im xml ein Element `invoice:tiers_garant` vorkommt, wird `DocAttr` auf `Tiers_Garant_Manuell` gesetzt. Wenn `DocAttr` nicht ermittelt werden kann wird `DocAttr` auf `direct` gesetzt.

Korrekte DocAttr Werte für EBPP und XML 4.3/4.4 Dokumente:		
Dokumenttyp	Bedingung	DocAttr
EbppInvoiceRequest_100.xsd		direct
EbppAdminMessage_100.xsd		direct
generalContainer_430.xsd	Enthält generalInvoiceRequest mit tiers_payant Element (mit oder ohne MCD)	Tiers_Payant
generalContainer_430.xsd	Enthält generalInvoiceRequest mit tiers_garant Element (mit oder ohne	Tiers_Garant

	MCD)	_Manuell
generalContainer_430.xsd	Enthält generalInvoiceRequest ohne tiers_payant oder tiers_garant Element (mit oder ohne MCD)	ERROR
generalContainer_430.xsd	Enthält generalInvoiceResponse (mit oder ohne MCD)	direct
generalContainer_430.xsd	Enthält nur MCD	direct
generalContainer_430.xsd	leer (weder invoice noch MCD)	ERROR
generalCreditRequest_430.xsd		direct
generalCreditResponse_430.xsd		direct
generalFormRequest_430.xsd		direct
generalFormResponse_430.xsd		direct
generalInvoiceRequest_[430 440].xsd	Enthält tiers_payant Element	Tiers_Payant
generalInvoiceRequest_[430 440].xsd	Enthält tiers_garant Element	Tiers_Garant _Manuell
generalInvoiceRequest_[430 440].xsd	Enthält weder tiers_payant noch tiers_garant Element	ERROR
generalInvoiceResponse_[430 440].xsd		direct
generalNotification_[430 440].xsd		direct
hospitalMCDRequest_[430 440].xsd		direct
hospitalMCDResponse_[430 440].xsd		direct
statusRequest_430.xsd		direct
statusResponse_430.xsd		direct

- TRUSTCENTEREAN, ISPAPERINVOICE, SCHEMAID: Da diese Parameter in der Steuerdatei ebenfalls fakultativ sind, werden sie weggelassen.

<b>mpcommunicator.config (optional)</b>		
<b>Key</b>	<b>Beispiel Value</b>	<b>Default</b>
CLIENT.n.DIR.m	data/client1/invoices1	
CLIENT.n.DIR.m.DOCATTR	Tiers_Payant	direct

CLIENT.n.DIR.m.DOCPRINTED	false	true
CLIENT.n.DIR.m.DISTTYPE	1	0
CLIENT.n.DIR.m.PRINTLANGUAGE	I	D
CLIENT.n.DIR.m.TRUSTCENTEREAN	1234567890123	
CLIENT.n.DIR.m.ISPAPERINVOICE	false	
CLIENT.n.DIR.m.SCHEMAID	26020	
CLIENT.n.DIR.m.PLUGIN	HL7	

Die übrigen in der Sende-Steuerdatei benötigten Parameter werden vom MPC automatisch vergeben:

- SenderDocId wird gemäss der im xml vorhandenen Referenz gesetzt, welche je nach Schema aus einem der folgenden Attribute gelesen wird: request\_id, invoice\_id, credit\_id, response\_id, id. Wenn keine Referenz vorhanden ist sowie bei Statusanfragen/-antworten, welche viele Referenzen enthalten können, wird SenderDocId wie FileName ohne Suffix gesetzt.
- Subject wird gleich wie SenderDocId gesetzt.

### 3.5.3 Konfiguration der Funktion

Die Funktion wird mit den folgenden Parametern in der Konfigurationsdatei `mpcommunicator.config` gesteuert:

#### Hinweis:

- `mpcommunicator.writesendcontrol.pollinterval` und `mpcommunicator.writesendcontrol.cron` sind analog zu den Standard-Funktionen zu spezifizieren und verhalten sich entsprechend (d.h. nur eine von beiden angeben, bzw. `mpcommunicator.writesendcontrol.cron` hat Vorrang)

<b>mpcommunicator.config (optional)</b>		
<b>Key</b>	<b>Beispiel Value</b>	<b>Default</b>
<code>mpcommunicator.task.writesendcontrol.enable</code>	true schaltet die Funktion ein, false aus.	false
<code>mpcommunicator.writesendcontrol.pollinterval</code>	Intervall der Ausführung dieser Funktion in Minuten.	5
<code>mpcommunicator.writesendcontrol.cron</code>	Cron für diese Funktion.	

Wenn die Funktion eingeschaltet ist, liest der MPC regelmässig alle mittels `CLIENT.n.DIR.m` spezifizierten Verzeichnisse, erstellt in den Sende-Verzeichnissen der zugehörigen Mandanten jeweils Sende-Steuerdateien gemäss den angegebenen Parameter-Sets und verschiebt die zu sendenden xmls in die Sende-Verzeichnisse der zugehörigen Mandanten. Dort werden die Dateien von der Sende-Funktion wie gewohnt weiter verarbeitet.

Bei RMI-Betrieb (Kap.0) kann die Funktion manuell mittels `MPC_RMI_Write_Send_Control_Start.exe` ausgeführt werden.

## 3.6 Archivierung

Ab Version 1.9.0 des MPC können gesendete xmls mit ihren zugehörigen Sende-Steuerdateien archiviert werden.

Pro Mandant kann in der Konfigurationsdatei ein Archivierungs-Verzeichnis spezifiziert werden. In diesem wird pro verarbeiteter Sende-Steuerdatei ein Verzeichnis `yyyy-MM-dd_HH-mm-ss_S` angelegt, welches den aktuellen Verarbeitungszeitpunkt in ms-Genauigkeit S kennzeichnet. In einem solchen Verzeichnis wird die entsprechende Sende-Steuerdatei mit allen darin erwähnten und erfolgreich versandten xmls abgelegt. Mit einer separaten Funktion werden Archive, die älter als eine vorgegebene Zeit sind, wieder gelöscht.

Wichtig: Wenn das Archivieren nicht möglich ist, wird das xml und die zugehörige Sende-Steuerdatei trotzdem gelöscht! Da beim Starten des MPCs geprüft wird, ob Archivieren möglich ist, sollte dieser Fall ohnehin nur auftreten können, wenn während der Laufzeit etwa die Schreibberechtigung fürs Archivieren entzogen wird.

mpcommunicator.config (optional)		
Key	Bedeutung	Default
CLIENT.n.SENDARCHIVE_DIR	Archivierungs-Verzeichnis	CLIENT.n.SEND_DIR/archive
mpcommunicator.task.archive.enable	true schaltet die Funktion ein, false aus	false
mpcommunicator.archivedelete.pollinterval	Intervall der Ausführung des Löschens archivierter xmls in Stunden	48
mpcommunicator.archivedelete.cron	Cron für das Löschen archivierter xmls	
mpcommunicator.archivedelete.maxtime	Anzahl Tage, bevor archivierte xmls gelöscht werden, wenn die Funktion eingeschaltet ist	365

## Hinweise zum Ablauf:

- Das Archivieren und das Löschen der Archive wird mit derselben Property ein- und ausgeschaltet. Ausgenommen hiervon ist das manuelle Löschen der Archive mit dem RMI-Kommando  
`MPC_RMI_Delete_Old_Archives_Start.exe`
- welche bei RMI-Betrieb immer funktioniert.
- Wenn eine Sende-Steuerdatei xmls enthält, von welchen einige erfolgreich und andere nicht erfolgreich gesendet wurden, wird die Sende-Steuerdatei sowohl in das Archiv-Verzeichnis (gemeinsam mit den erfolgreich gesendeten xmls) als auch in das Fehler-Verzeichnis (gemeinsam mit den nicht erfolgreich gesendeten xmls) gelegt.
- Wenn der MPC bei einer erst teilweise verarbeiteten Sende-Steuerdatei gestoppt wird und später wieder gestartet wird, wird nach dem Neustart ein neues Archivverzeichnis für den nicht-verarbeiteten Teil der Sende-Steuerdatei erstellt, und diese wird sowohl im ersten wie auch im neuen Archivverzeichnis abgelegt.
- Wenn ein Archivverzeichnis keine erfolgreich versandten xmls enthält, wird es automatisch wieder gelöscht.

Wie die Sende-Verzeichnisse können auch die Archiv-Verzeichnisse mit dem `smb://` Protokoll konfiguriert werden (da sie defaultmässig Unterverzeichnisse der Ersteren sind) (Kap. 5.3.3). Wichtig: Wenn das `smb://` Protokoll eingesetzt werden soll, müssen immer alle Verzeichnisse, welche für `smb://` konfigurierbar sind, dafür konfiguriert werden, nicht nur einige hiervon.

## 3.7 Generische Funktionen

### 3.7.1 Funktionsbeschreibung

Ab Version 1.9.0 des MPC können eine beliebige Anzahl beliebiger selbst erstellter Funktionen vom MPC ausgeführt werden. Mit dieser "Plug-In"-Funktionalität können z.B. aus proprietären Daten XML-Standardrechnungen für den Versand mit dem MPC aufbereitet werden, es können einfach SendControl-Dateien erstellt werden, empfangene Rückweisungen können weiterverarbeitet werden u.v.a.m.

Generische Funktionen können wie die eingebauten Standard-Funktionen mittels Intervallscheduler, mittels cron-Scheduler sowie ebenfalls beim MPC-Betrieb als Windows Service genutzt werden, jedoch ist ein manueller Betrieb via RMI z.Zt. nicht vorgesehen. Ferner können generische Funktionen im MPC nicht mandanten-abhängig konfiguriert werden (ggf. kann man dies ja problemlos in der Funktion selber implementieren).

### 3.7.2 Implementierung von generischen Funktionen

Die kontrollierende Java-Klasse muss `SkipableJob` erweitern (enthalten in `MPCCommunicator/lib/MPCCommunicator.jar`) und die Methode `execute_internal()` implementieren. Diese wird dann gemäss Konfiguration vom Task-Scheduler im MPC ausgeführt:

```
public class MyPlugin extends
com.medidata.mediport.communicator.scheduler.SkipableJob
public void execute_internal() { ... }
```

#### **Hinweis:**

Der Java-source-level der hinzugefügten Klassen muss mit der JVM Version kompatibel sein. Defaultmässig wird für die meisten Betriebssysteme eine JVM Version 1.6 im Verzeichnis `MPCCommunicator/jre/` installiert. Diese kann natürlich mit einer anderen JVM ersetzt werden, oder Pfade können entsprechend angepasst werden.

### 3.7.3 Konfiguration von generischen Funktionen

In der Konfigurationsdatei `MPCCommunicator/config/mpcommunicator.config` müssen je generischer Funktion entsprechende Properties gesetzt werden.

## Hinweis:

- *n* steht für die Nummer der Funktion. Diese Nummerierung soll bei 1 beginnen und bei mehreren Funktionen fortlaufend (1,2,3,...) erfolgen.
- `extension.n.pollinterval` und `extension.n.cron` sind analog zu den Standard-Funktionen zu spezifizieren und verhalten sich entsprechend (d.h. nur eine von beiden angeben, bzw. `extension.n.cron` hat Vorrang).
- Konfigurierte generische Funktionen sind automatisch eingeschaltet (d.h. zum Ausschalten kann der `extension.n.job` Parameter auskommentiert werden).

mpcommunicator.config (optional)			
Key	Bedeutung	Beispiel	Default
<code>extension.n.job</code>	Name der Java-Klasse, welcher periodisch die Kontrolle übergeben wird.	<code>ch.md.mp.mpc.ext.mycompany.MyPlugin</code>	
<code>extension.n.pollinterval</code>	Intervall der Ausführung dieser Funktion in Minuten.	5	10
<code>extension.n.cron</code>	Cron für diese Funktion.		

Ferner müssen alle Pfade (java classpaths) aller verwendeten Start-Dateien mit den jars/classes der neu hinzugefügten Funktionen ergänzt werden:

- Linux: **MPCCommunicator/MPC\_Scheduler\_Start.lax** anpassen (`lax.class.path` ergänzen)
- Windows: **MPCCommunicator/MPC\_Scheduler\_Start.lax** anpassen (`lax.class.path` ergänzen), bei Konsolenbetrieb **MPCCommunicator/bin/StartMPCCommunicator.bat** anpassen, für Windows Service Betrieb **MPCCommunicator/config/wrapper.conf** anpassen (`wrapper.java.classpath.n` hinzufügen)

## 3.8 Multi User Funktionalität

### 3.8.1 Funktionsbeschreibung

Ab Version 1.5.0 des MPC gibt es eine Mehrbenutzer -Unterstützung. Sämtliche Standard-Funktionen (Senden, Empfangen, DokStatus, Partnerverzeichnis) können für jeden Benutzer einzeln konfiguriert werden. Ab Version 1.9.0 kann eine beliebige Anzahl Benutzer mittels beliebigen (positiven) Integer-Zahlen spezifiziert werden (statt wie bisher `CLIENT.1.EAN=...` anzugeben kann z.B. `CLIENT.1000000.EAN=...`, dann `CLIENT.2000000.EAN=...` etc. konfiguriert werden - alle Konfigurationsparameter mit derselben Nummer *n* in `CLIENT.n.XYZ` werden demselben `CLIENT` zugerechnet - der frühere Parameter `CLIENTS.MAXNUMBER` wird nicht mehr benötigt). Die Benutzer werden dabei pro Task nacheinander abgearbeitet.

## Hinweis:

- Die verschiedenen EAN Nummern der einzelnen Clients müssen für das vom MPC verwendete SSL Zertifikat (Keystore) von der MediData freigeschaltet worden sein.
- Ab MPC Version 1.9.0 dürfen verschiedene Clients auch gemeinsame Sendeverzeichnisse verwenden.
- Die Abrechnung erfolgt pro Client (resp. pro EAN Nummer)

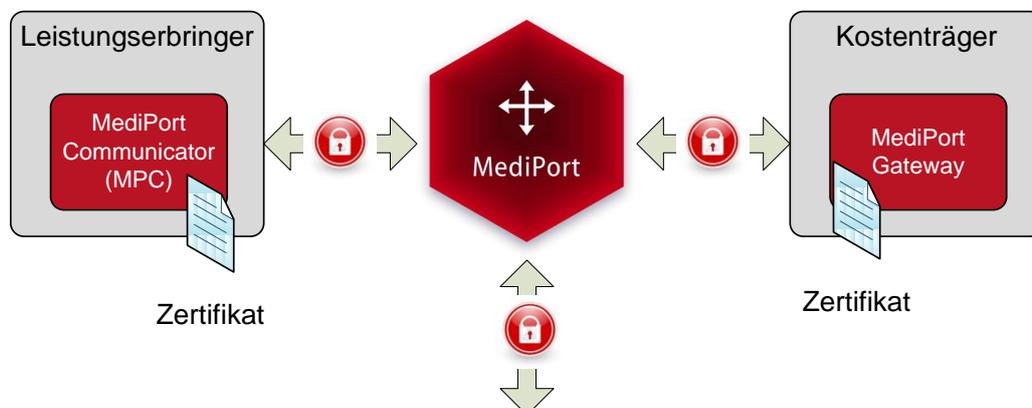
## 3.9 Sicherheitsfunktionen / Zertifikate

- Es gelangt ein Zertifikatsspeicher in Form einer verschlüsselten Datei zum Einsatz
- Der Zertifikatsspeicher enthält die Zertifikate des aktuellen Jahres und die Passwörter zum Lesen der Zertifikate
- Ein Wechsel der Zertifikatsspeicher muss Anfangs Jahr geschehen

Zu den Sicherheitsfunktionen gehören alle Funktionen im Zusammenhang mit der **sicheren** Übertragung von Dokumenten von und zu MediPort. Dazu gehören Authentifizierung, Verschlüsselung und Zertifikatshandling.

### 3.9.1 Authentifizierung und Verschlüsselung

Verbindungen mit dem MediPort-Server erfolgen immer unter Verwendung des SSL - Verfahrens mit Authentifizierung von **Client und Server** und Verschlüsselung der Daten während der Übertragung. Diese Technologie stellt sicher, dass es sich bei den Kommunikationspartnern um diejenigen handelt, die sie vorgeben zu sein und dass die Daten von Unbeteiligten nicht eingesehen werden können.



## 3.9.2 Zertifikatshandling

Für die Kommunikation mit MediPort liefert MediData AG jeweils eine CD mit einem Schlüsselspeicher (Keystore) der alle gültigen Zertifikaten des entsprechenden Jahres enthält.

## 3.10 Prozesssteuerung

### 3.10.1 Timer / Taskmanager

- Ein Timer steuert die Auslösung der drei verschiedenen Tasks:
  - senden von Daten
  - empfangen von Daten
  - update des Partnerverzeichnisses
- Die Zeitintervalle werden in der Konfigurationsdatei definiert
- Tasks werden nicht ausgeführt, wenn ein gleichartiger Task noch am Ausführen ist.
- Bei Kommunikationsproblemen mit dem MediPort Server wird die Ausführung des Tasks abgebrochen und beim nächsten definierten Zeitpunkt wieder fortgesetzt.

### 3.10.2 Externe Trigger

Der MPC kann anstelle der Timer- / Tasksteuerung auch über externe Trigger aktiviert werden.

## 3.11 Protokollierung

### 3.11.1 Aufbau der Logmeldungen

- Eine Log-Datei ist die Schnittstelle zum Anwender bzw. Anwenderprogramm
- Der Log-Level kann von INFO bis FATAL ERROR eingestellt werden
- Ein Log-Datei Eintrag ist folgendermassen aufgebaut:
  - Zeitstempel (23 Zeichen)
  - Leerzeichen (1Zeichen)
  - NDC (nested diagnostic context) pro Task (S=Send / R= Empfang / U=PartnerUpdate / M=Haupttask / F=FileStop / C=ConsoleStop) (1Zeichen)
  - Leerzeichen (1Zeichen)
  - Log-Level (5 Zeichen, aufgefüllt mit Leerzeichen))
  - Leerzeichen (1Zeichen)
  - (1 Zeichen)
  - Leerzeichen (1Zeichen)
  - Eindeutig identifizierbare Server-Meldungen (Siehe Anhang B)

#### Beispiel eines Logfileeintrages:

```
2003-03-12 09:49:26,906 P INFO - I2902: ADRBOOK UNCHANGED
```

Ein ausführlicheres Beispiel eines MP Communicator Logfiles befindet sich im Anhang. Die Konfiguration ist beschrieben im Kapitel „Konfiguration“.

## 4. Installation

### 4.1 Allgemein

#### 4.1.1 Windows 10 / 8 / 7

Bei einer Windows-Installation muss besonders beachtet werden, dass seit Windows Vista Programm-Installationsdateien und Benutzerdateien je nach Benutzer i.d.R. nicht mehr denselben Pfad verwenden dürfen. Die Applikation selber wird in ein Verzeichnis <INSTALLDIR> installiert, typischerweise etwa C:\Program Files\MPCCommunicator\, dieses Verzeichnis bleibt jedoch normalerweise für Benutzer ohne Administratoren-Rechte schreibgeschützt. Damit nicht-Administratoren danach den MPCCommunicator auch betreiben können, wird ein separates Konfigurations- und Datenverzeichnis <DATADIR> wie z.B. D:\mpcdata\ verwendet, wo alle vom Benutzer konfigurierbaren sowie verwendeten Dateien sowie die gesendeten/empfangenen Xmls etc. abzulegen sind. Da der MPCCommunicator von verschiedenen Benutzerkonten aus konfiguriert und betrieben werden kann, wird der Übersichtlichkeit halber ein einziges benutzerunabhängiges <DATADIR> verwendet. Wenn für <DATADIR> ein Verzeichnis verwendet würde, für das der aktuelle Benutzer nicht genügend Rechte hätte, würden alle Schreibzugriffe von Windows automatisch in ein anderes, sog. "VirtualStore" Verzeichnis "umgebogen", was zu grosser Verwirrung führen könnte. Aus diesem Grund soll bei der Installation ein vom Betriebssystem nicht besonders geschütztes Verzeichnis wie etwa D:\mpcdata\ als <DATADIR> gewählt werden, für welches alle potentiellen Betreiber-Konten Schreibberechtigung haben. Die Installation selber muss üblicherweise als Administrator ausgeführt werden (rechte Maustaste-"Als Administrator ausführen").

#### 4.1.2 Erstinstallation

##### Starten des Installers

- MediPort\_Communicator\_VXXX.exe [Windows]
- MediPort\_Communicator\_VXXX.dmg [Mac OS X]
- MediPort\_Communicator\_VXXX.sh [Linux]

direkt von der Installations-CD im Verzeichnis „installer“. Anstelle des GUI Installers kann der Installer mittels der Option „-c“ auch von der Konsole aus gestartet werden, oder mit der Option "-q" im unbeaufsichtigten Modus, wo bei allen Eingaben automatisch Default-Werte gewählt werden. Es muss ein Programm-Installationsverzeichnis <INSTALLDIR> sowie ein Daten-Installationsverzeichnis <DATADIR> gewählt werden. Ausser in neueren Windows-Versionen (s.o.) kann hierfür auch ein- und dasselbe Verzeichnis gewählt werden.

##### Hinweise:

- Wichtig: <DATADIR> sollte unbedingt auf einem lokalen Laufwerk gewählt werden, sonst können bei einem Netzwerkunterbruch weder das Log noch manche Systemdateien geschrieben werden und das Problem könnte nicht erkannt werden!

- Neues Keystorefile (befindet sich auf der Zertifikats CD, wird separat geliefert) zur Verfügung stellen:
  - Kopieren des Keystorefiles nach <DATADIR>/config/mpcommunicator.keystore
- Konfigurieren Sie den MPC entsprechend den Hinweisen im Kapitel Konfiguration.

### 4.1.3 Verzeichnisstruktur nach Installation

<INSTALLDIR> z.B. ...MPCCommunicator/

- bin : Enthält Dateien zum starten/stoppen/testen der Applikation und zum Installieren/Deinstallieren als Service (nur Windows)
- config : Enthält fixe und default Konfigurationsdateien
- Documentation : Enthält die Produktdokumentation
- jre : Enthält die Java Laufzeitumgebung
- lib : Enthält alle benötigten Laufzeitbibliotheken
- xsd: Schema-Dateien
- Wrapper.exe (nur Windows): Windows Service Wrapper
- MPC\_xy Start und MPC\_xy Stop Programme

<DATADIR> z.B. /opt/mpcdata/ oder D:\mpcdata\

- config: enthält benutzerkonfigurierbare Konfigurationsdateien und Zertifikats-Keystore.
- data : Defaultmässig enthält dieses Verzeichnis alle Dateien die vom MP Communicator verarbeitet werden (Send/Receive/Partnerverzeichnis) sowie die dabei erzeugten Logfiles
  - send : Enthält die zu versendenden Daten
    - archive: Enthält archivierte gesendete Daten
  - receive : Enthält die empfangenen, produktiven Daten
    - test : Enthält die empfangenen Testdaten
  - error : Enthält die Daten mit einem Sendfehler
  - log : Enthält die Logdatei und ihre Backups.
  - partner : Enthält das Partnerverzeichnis.
  - docstatus : Enthält die Listen mit den Dokumentstati.

## 4.2 Windows spezifisch

Bitte beachten Sie den allgemeinen Teil der Installation.

### 4.2.1 Installation als Windows Service (Optional)

Ausführen von <INSTALLDIR>/bin/InstallMPCCommunicatorServiceWrapper-Win.bat

Wichtig:In neueren Windows-Versionen wie Windows 8 und 10 können Dienste üblicherweise nur vom Administrator installiert werden. Hierfür die Datei mit der rechten Maustaste "Als Administrator ausführen" ausführen.

#### Hinweis:

- Der Service startet erst beim nächsten Start von Windows. Alternativ kann der Service über das Windows Dienste Menü von Hand gestartet werden. Der Service hat den Name „MediData MediPort Communicator“.
- Der für den Service benutzte User muss die konfigurierten Pfade (z.B Laufwerk mappings) bereits haben.

## 4.3 Linux / Unix spezifisch

Bitte beachten Sie den allgemeinen Teil der Installation.

### 4.3.1 Das Linux / Unix Demon Startskript mpc.sh

Dieses Skript unterstützt folgende Optionen:

- start: Startet den MPC als Demon
- stop: Stoppt den MPC Demon
- console: Startet den MPC in der momentanen Shell
- restart: Startet den MPC Demon neu.
- status: Gibt den Status des MPC zurück

### 4.3.2 Installation als Linux / Unix Daemon (Optional)

Erzeugen eines symbolischen Links ins /etc/init.d Verzeichnis (als root und mit dem ganzen Pfad):

```
ln -s <INSTALLDIR>/mpc.sh /etc/init.d/mpcommunicator
```

Erzeugen der symbolischen Links in die einzelnen Runlevels:

Entweder mit update-rc.d mpcommunicator start 20 2 3 4 5 . Stopp 20 0 10 6 . oder sonst von hand:

```
ln -s /etc/init.d/mpcommunicator /etc/rc0.d/K20mpcommunicator
ln -s /etc/init.d/mpcommunicator /etc/rc1.d/K20mpcommunicator
ln -s /etc/init.d/mpcommunicator /etc/rc6.d/K20mpcommunicator
```

```
ln -s /etc/init.d/mpcommunicator /etc/rc2.d/S20mpcommunicator
ln -s /etc/init.d/mpcommunicator /etc/rc3.d/S20mpcommunicator
ln -s /etc/init.d/mpcommunicator /etc/rc4.d/S20mpcommunicator
ln -s /etc/init.d/mpcommunicator /etc/rc5.d/S20mpcommunicator
```

### Hinweise:

Sollte beim Starten des Daemons folgende Meldung erscheinen

```
/etc/init.d/mpcommunicator: /home/support/MPCCommunicator/wrapper: /lib/ld-linux.so.2: bad ELF interpreter: Datei oder Verzeichnis nicht gefunden
```

bedeutet dies, dass 32-bit Libraries benötigt werden, die nicht vorhanden sind.

Die Libraries können manuell installiert werden.

```
yum -y install glibc.i686
```

Wenn unter Ubuntu beim Versuch, den Service zu starten, folgende Meldung erscheint

```
/etc/init.d/mpc: 192: exec: ./wrapper: not found
```

ist ebenfalls die Ursache, dass für das Programm, welches den Service startet, 32-bit Libraries benötigt werden.

Dies kann mit der Installation von 32bit-Libraries gelöst werden:

```
sudo dpkg --add-architecture i386  
sudo apt-get update  
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386
```

## 5. Konfiguration

### 5.1 Minimale Konfiguration

Bei einer Installation des MediPort Communicators müssen folgende Anpassungen an den Defaulteinstellungen im Konfigurationsfile <DATADIR>/config/ mpcommunicator.config gemacht werden:

mpcommunicator.config (minimal)		
Key	Bedeutung	Beschreibung
mpcommunicator.sender.ean CLIENT.1.EAN	MediPort Communicator Sender EAN	Die hierfür nötigen Daten befinden sich auf der Installations-CD im Zertifikat Ordner in der Datei xyz.re.
mediport.dn	MediPort Communicator Sender DN	Die hierfür nötigen Daten befinden sich auf der Installations-CD im Zertifikat Ordner in der Datei xyz.re
mpcommunicator.mediport.ip	MediPort Host Name or Host Address	Je nach Testbetrieb / Produktivbetrieb muss die IP Adresse des MediPort Host gewechselt werden.  <b>Produktiver Server:</b> mpw1.medidata.ch <b>Test Server:</b> mpw0.medidata.ch

#### Hinweis:

```
#=====
# MINIMAL CONFIGURATION
#=====

# MediPort Communicator Sender EAN
mpcommunicator.sender.ean=2099988870017

# MediPort Communicator Sender DN
mediport.dn=uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch

# MediPort Host Name or Host Address
# Produktiver Server: mpw1.medidata.ch
# Test Server : mpw0.medidata.ch
mpcommunicator.mediport.ip=mpw0.medidata.ch

CLIENT.1.EAN=2099988870017
```

Alle Sende- und Empfangsverzeichnisse sind korrekt gesetzt und müssen nicht mehr angepasst werden.

## 5.2 Optionale Konfiguration

Folgende Konfigurationseinstellungen können optional in der Konfigurations-datei <DATADIR>/config/mpcommunicator.config durchgeführt werden:

### 5.2.1 Intervallscheduler

Minimale Werte für die Intervalle sind:

- PartnerUpdate: 60 [Minuten]
- Send: 5 [Sekunden]
- Receive: 10 [Minuten]
- getdocstate: 24 [Stunden]

#### Hinweis:

- Falls diese Minimalwerte in der Konfiguration unterschritten werden, werden automatisch diese Minimalwerte benutzt.
- Der Intervallscheduler ist nur aktiv wenn in der Konfigurationsdatei „rmi.enable=false“ gesetzt ist (Default Einstellung).

Intervallscheduler mpcommunicator.config (optional)		
Key	Bedeutung	Default
mpcommunicator.partner.updateinterval	MPC Partnerverzeichnis update Intervall in Minuten.	240
mpcommunicator.send.pollinterval	MPC Send poll Intervall in Sekunden.	10
mpcommunicator.receive.pollinterval	MPC Receive poll Intervall in Minuten.	60
mpcommunicator.getdocstate.pollinterval	MPC Getdocstate poll Intervall in Stunden.	24

## Beispiel:

```
#####
# TASKS (SIMPLE)
#####
# REMARK: Minimal values for the intervals are:
#   - PartnerUpdate : 60 [minutes]
#   - Send          : 5 [seconds]
#   - Receive       : 10 [minutes]
#   - Getdocstate: 24 [hours]
#####

# MediPort Communicator Partnerverzeichnis update interval[m]
mpcommunicator.partner.updateinterval=240

# MediPort Communicator Send poll interval[s]
mpcommunicator.send.pollinterval=10

# MediPort Communicator receive poll interval[m]
mpcommunicator.receive.pollinterval=60

# MediPort Communicator getdocstate poll interval[h]
mpcommunicator.getdocstate.pollinterval=24
```

### 5.2.2 Prüfungen vor dem Senden

Diese Properties sollten im Normalbetrieb nicht benötigt oder verändert werden, weshalb sie standardmässig in der mpcommunicator.config Datei weggelassen wurden. Wenn sie nicht explizit gesetzt sind, werden automatisch die Default-Werte verwendet.

Senden-Prüfungen mpcommunicator.config (optional)		
Key	Bedeutung	Default
schemadir	Verzeichnis mit Schema-Dateien	<INSTALL DIR>/xsd/
maxvalidationlogentries	max. Anzahl Validerungsfehler im Log pro Xml	5
mpcommunicator.send.checkunique.enable	Mehrfaches Senden derselben Datei prüfen ein/ausschalten (true/false)	true
mpcommunicator.send.checkunique.interval	Dauer während der ein Xml gespeichert wird für das Prüfen mehrfach gesendeter Xmls, in Minuten	60
uniquedocs	max. Anzahl der zuletzt gesendeten Xmls, die für das Prüfen mehrfach gesendeter Xmls verwendet werden; wenn innerhalb der letzten Stunde weniger Xmls versandt wurden, werden nur jene geprüft	5000
maxhashsize	max. Grösse in KB eines Xmls, bei der das gesamte Dokument mit bereits gesendeten verglichen wird; bei grösseren Xmls wird nur die SenderDoc ID verglichen	200

Senden-Prüfungen mpcommunicator.config (optional)		
deletespercheckpoint	Anzahl Xmls, nach welcher die interne Datenbank "aufgeräumt" wird	5000
mpcommunicator.send.checkintermediate.enable	Intermediär eines zu sendenden Xmls prüfen ein/ausschalten (true/false)	true
mpcommunicator.send.fixintermediate.enable	Intermediär eines zu sendenden Xmls falls anders zu MediData EAN korrigieren ein/ausschalten (true/false) (Voraussetzung: Prüfen des Intermediärs eingeschaltet)	false
mpcommunicator.send.check.addrbook	Senden verhindern falls PartnerVerzeichnis nicht aktuell ist.	true

## 5.2.3 Sendcontrol

Hier wird der Prefix (Die Namen der von Ihnen verwendeten Steuerdateien müssen mit dieser Zeichenkette beginnen) der Sendcontrol Datei bestimmt. Die Sendcontrol datei enthält die Informationen zu den zu versendeten Dateien. Die Sendcontrol Datei muss sich im entsprechenden Send Verzeichnis befinden

### Hinweis:

- Je nach verwendetem Betriebssystem ist der Name case sensitiv (Es wird also zwischen Gross- und Kleinschreibweise unterschieden).
- Die effektiv erzeugte Sendcontrol Datei kann nach dem Prefix noch eine beliebige Zeichenkette enthalten, muss aber mit „.xml“ enden (Postfix).

Sendcontrol mpcommunicator.config (optional)		
Key	Bedeutung	Default
mpcommunicator.sender.controlfile	Sender Steuerdatei (Case sensitive). Die Namen der von Ihnen verwendeten Steuerdateien müssen mit dieser Zeichenkette beginnen.	SendControl

### Beispiel:

```
#####
# SENDCONTROL
#####
# MediPort Communicator Sender Steuerdatei (SendControl010_xyz.xml)
```

```
# REMARK: !!! Case sensitive!!! SendControl010_ != sendcontrol010_  
mpcommunicator.sender.controlfile=SendControl
```

## 5.2.4 Schemaprüfung und Antwortversand beim Empfang

Für die Konfiguration der Schemaprüfung und Antwortversand beim Empfang ist ein separates Konfigurationsfile notwendig namens `receive_validator.config`, welches im gleichen Verzeichnis wie `mpcommunicator.config` abzulegen ist.

**Wichtig:** Alle Konfigurationen in `receive_validator.config` sind mandantenspezifisch, d.h. für jede GLN-Nummer eines Mandanten muss ein eigenes Set von Properties konfiguriert werden, die jeweils mit der entsprechenden GLN-Nummer beginnen.

Die bei der Installation mitgelieferte Version von `receive_validator.config` zeigt alle möglichen Konfigurationsproperties. Die Schemaprüfung kann sowohl global für alle Schemas als auch pro Schema einzeln ein/ausgeschaltet werden, indem `true` oder `false` gesetzt wird. Die schemaspezifischen Properties sind optional und können mit `#` auskommentiert oder weggelassen werden. In diesem Fall gilt einfach die globale Property für diesen Mandanten. Damit die Schemaprüfung für einen Dokumenttyp X stattfindet, darf weder die globale noch eine allfällige schemaspezifische Property `false` sein. Wenn `receive_validator.config` nicht vorhanden ist, wird nichts gemacht.

Ob eine automatische Antwort generiert wird kann nur global für alle Schemas ein/ausgeschaltet werden. Für diese Antwort ist die Bezeichnung und Version des Antwortgenerators sowie die in der Antwort abzufüllenden Kontaktdaten zu konfigurieren.

Die Property `xml.generator.pretty_print` schaltet die Verwendung von xml pretty print in der generierten Antwort ein/aus und muss in `mpcommunicator.config` konfiguriert werden:

mpcommunicator.config (optional)		
Key	Bedeutung	Default
<code>xml.generator.pretty_print</code>	Verwendung von xml pretty print in modifizierten empfangenen sowie generierten Dokumenten ein/ausschalten (true/false)	false

## 5.2.5 Überschreiben von recipient und insurance beim Empfang

Für die Konfiguration des Überschreibens von recipient und insurance beim Empfang ist ein separates Konfigurationsfile notwendig namens `receive_modifier.config`, welches im gleichen Verzeichnis wie `mpcommunicator.config` abzulegen ist.

**Wichtig:** Alle Konfigurationen in `receive_modifier.config` sind mandantenspezifisch, d.h. für jede GLN-Nummer eines Mandanten muss ein eigenes Set von Properties konfiguriert werden, die jeweils mit der entsprechenden GLN-Nummer beginnen.

Die bei der Installation mitgelieferte Version von receive\_modifier.config zeigt alle möglichen Konfigurationsproperties. Das Überschreiben von recipient-GLN sowie der insurance-Daten kann sowohl global für alle Schemas als auch pro Schema einzeln ein/ausgeschaltet werden, indem true oder false gesetzt wird. Wenn die insurance-Daten angepasst werden sollen, müssen auch immer alle 5 dazugehörigen Properties konfiguriert sein, sonst werden sie nicht verändert.

Die Property xml.generator.pretty\_print schaltet die Verwendung von xml pretty print im empfangenen Dokument ein/aus wenn dessen recipient oder insurance Daten überschrieben werden. Diese Property muss in mpcommunicator.config konfiguriert werden (vgl. Definition oben).

## 5.2.6 Prozesssteuerung Trigger (RMI)

Diese Konfiguration wird benötigt um den MPC nicht über den eingebauten Scheduler sondern von extern mit Triggern zu steuern.

### Hinweis:

Der angegebene RMI Port wird ab V1.4.0 in jedem Fall belegt um ein Mehrfachstarten derselben MPC Instanz zu unterbinden.

Prozesssteuerung Trigger (RMI) mpcommunicator.config (optional)		
Key	Bedeutung	Default
rmi.serverip	IP des Rechners auf dem der MPC läuft. Hier kann praktisch immer „localhost“ eingetragen werden.	localhost
rmi.serverport	Dieser Port des Rechners wird vom MPC beim Aufstarten belegt.	4445
rmi.servicename	RMI ServiceName. Sollte nicht geändert werden.	MPCCommunicator-Server
rmi.enable	Schalten den Trigger Prozesssteuerung ein und damit automatisch die Scheduler Steuerung aus.	false

### Beispiel:

```
#####
# RMI Server
#####
# REMARK: If rmi.enable is set to true then all the
#   tasks are set to false and you can only execute
#   a task by calling its rmi interface.
#####
rmi.serverip=localhost
rmi.serverport=4445
rmi.servicename=MPCCommunicator-Server
```

rmi.enable=false

## 5.2.7 Proxykonfiguration

Falls Sie keinen direkten Internetzugang haben sondern nur über einen (http) Proxy Server mit dem Internet Verbindung haben, können Sie diesen Proxy Server hier konfigurieren.

Proxykonfiguration mpcommunicator.config (optional)		
Key	Bedeutung	Default
proxy.use	Use Proxy Server (true/false).	false
proxy.ip	Proxy Server Host Name or Host Address	-
proxy.port	Proxy Server Port	-
proxy.auth.use	Use Proxy Server User/Password authentication (true/false). Benutzt die Proxy Server Authentifikation mit den unten angegebenen Usernamen und Passwörter.	false
proxy.auth.username	Proxy Server Authentifikation Username	-
proxy.auth.password	Proxy Server Authentifikation Password	-

### Beispiel:

```
#####
# Proxy Server
#####
# REMARK: Only "Basic" Proxy Authentication is supported
#####

# Use Proxy Server (true/false)
proxy.use=false

# Proxy Server Host Name or Host Address (192.168.1.122/192.168.3.51)
proxy.ip=1.1.1.1

# Proxy Server Port(8443/8100)
proxy.port=8443

# Use Proxy Server User/Password authentication (true/false)
proxy.auth.use=true

# Proxy Server authentication username
proxy.auth.username=name

# Proxy Server authentication password
proxy.auth.password=password
```

## 5.2.8 Pfadkonfiguration

Die Pfadkonfiguration wurde ab MPC V1.5.0 geändert.

### Hinweis:

- Die entsprechenden Verzeichnisse müssen angelegt werden oder der MPC muss die Berechtigung haben, um neue Verzeichnisse zu erzeugen. Achtung: nicht gemachte Laufwerk mappings können nicht durch den MPC erzeugt werden.
- Relative Pfadangaben (default) werden als unterhalb von <DATADIR> (Kap. 4) interpretiert. Absolute Pfadangaben müssen mit / (Unix-Systeme) bzw. *Laufwerksbuchstabe*:, \\ oder smb:// (Windows) beginnen. Bei Verwendung von "\" im Pfad muss dieses Zeichen immer doppelt angegeben werden. Wenn z.B. auf einem Windows-Gerät <DATADIR> als "D:\mpcdata" gewählt wurde, sind die folgenden 3 Konfigurationen identisch:
  - CLIENT.1.SEND\_DIR=data/send
  - CLIENT.1.SEND\_DIR=D:/mpcdata/data/send
  - CLIENT.1.SEND\_DIR=D:\mpcdata\data\send
- x steht für beliebige (positive) Integer-Zahlen. Alle Parameter mit demselben x werden demselben Mandanten zugewiesen.

### Obsolet:

- Die Konfiguration der MPC vor V1.5.0 funktionieren momentan noch weiter, werden aber eventuell in einer späteren Version nicht mehr unterstützt.

Pfadkonfiguration mpcommunicator.config (optional)		
Key	Bedeutung	Default
CLIENT.x.EAN	EAN des jeweiligen Clients. Diese EAN muss für das verwendete SSL Zertifikat (Keystore) von der MediData freigeschaltet worden sein.	
CLIENT.x.SEND_DIR	Sendeverzeichnis des jeweiligen Clients.	
CLIENT.x.RECEIVE_DIR	Empfangsverzeichnis für produktive Dokumente des jeweiligen Clients.	
CLIENT.x.RECEIVETEST_DIR	Empfangsverzeichnis für Testdokumente des jeweiligen Clients.	
CLIENT.x.ERROR_DIR	Errorverzeichnis des jeweiligen Clients.	
CLIENT.x.DOCSTAT_DIR	Dokumentenstatus Verzeichnis des jeweiligen Clients.	
CLIENT.x.PARTNER_FILE	Partnerverzeichnis Datei des jeweiligen Clients.	

## Beispiel (Normalfall 1 Client):

```
#####  
# PFADKONFIGURATION  
#####  
  
### NEUE MULTI CLIENT (SENDER) AB V01.05.00  
# CLIENT 1 (of 100)  
CLIENT.1.EAN=2099988870017  
CLIENT.1.SEND_DIR=data/send  
CLIENT.1.RECEIVE_DIR=data/receive  
CLIENT.1.RECEIVETEST_DIR=data/receive/test  
CLIENT.1.ERROR_DIR=data/error  
CLIENT.1.DOCSTAT_DIR=data/docstatus  
CLIENT.1.PARTNER_FILE=data/partner/partnerinfo.txt
```

## Beispiel (Sonderfall 2 Clients):

```
#####  
# PFADKONFIGURATION  
#####  
  
### NEUE MULTI CLIENT (SENDER) AB V01.07.00  
  
# CLIENT 1  
CLIENT.1.EAN=2099988870017  
CLIENT.1.SEND_DIR=data/send  
CLIENT.1.RECEIVE_DIR=data/receive  
CLIENT.1.RECEIVETEST_DIR=data/receive/test  
CLIENT.1.ERROR_DIR=data/error  
CLIENT.1.DOCSTAT_DIR=data/docstatus  
CLIENT.1.PARTNER_FILE=data/partner/partnerinfo.txt  
  
# CLIENT 2  
CLIENT.2.EAN=2099988870017  
CLIENT.2.SEND_DIR=data2/send  
CLIENT.2.RECEIVE_DIR=data2/receive  
CLIENT.2.RECEIVETEST_DIR=data2/receive/test  
CLIENT.2.ERROR_DIR=data2/error  
CLIENT.2.DOCSTAT_DIR=data2/docstatus  
CLIENT.2.PARTNER_FILE=data2/partner/partnerinfo.txt
```

### 5.2.9 Dokumentenstatus

Über diese Funktion können die Stati der gesendeten Dokumente abgefragt werden. Um diese Funktion zu deaktivieren muss der getdocstate Task abgeschaltet werden (mpcommunicator.task.getdocstate.enable=false).

#### Hinweis:

- Der Task ist defaultmässig deaktiviert.
- Die Zeit der letzten Statusabfrage wird in einer Datei (DOCS\_STATE\_OUTPUTDIR/getdocstate.properties) zwischengespeichert.

Dokumentenstatus mpcommunicator.config (optional)		
Key	Bedeutung	Default
DOCS_STATE_INITIAL_DAYS	Beim Ausführen des getdocstate Tasks werden alle Dokumente die ihren Status in den letzten DOCS_STATE_INITIAL_DAYS Tagen oder seit dem letzten Aufruf geändert haben protokolliert.	100

## Beispiel:

```
#####
# DOCS_STATE
#####

# get all the docs which are done since today - DOCS_STATE_INITIAL_DAYS or
# since the last time the process was started
DOCS_STATE_INITIAL_DAYS=100
```

## 5.3 Advanced Konfiguration

### 5.3.1 Cronscheduler

Anstelle des normalen Intervallschedulers kann auch der Cronscheduler benutzt werden. Mit Hilfe dieses Schedulers kann sehr genau festgelegt werden, zu welchem Zeitpunkt der entsprechende Task ausgeführt werden soll. Diese Trigger beherrschen die komplette Unix Cron Syntax (Siehe Anhang C). Sobald der entsprechende CronTrigger definiert ist, wird der normale Trigger deaktiviert. Für folgende Task können im Konfigurationsfile <DATADIR>/config/mpcommunicator.config CronTrigger definiert werden:

Cronscheduler mpcommunicator.config (advanced)		
Key	Bedeutung	Default
mpcommunicator.partner.update.cron	MPC Partnerverzeichnis update Cron	-
mpcommunicator.send.cron	MPC Send Cron	-
mpcommunicator.receive.cron	MPC Receive Cron	-
mpcommunicator.getdocstate.cron	MPC Getdocstate Cron	

## Beispiel:

```
# MediPort Communicator Partnerverzeichnis update cron
mpcommunicator.partner.update.cron=0 0/1 10-13 ? * WED,FRI
```

```
# MediPort Communicator Send cron
mpcommunicator.send.cron=0 0/1 10-13 ? * WED,FRI

# MediPort Communicator receive cron
mpcommunicator.receive.cron=0 0/1 10-13 ? * WED,FRI

# MediPort Communicator getdocstate cron
mpcommunicator.getdocstate.cron=0 0 10 ? * WED,FRI
```

## 5.3.2 Mehrere MPC Services

Damit der MPC unter Windows mehrfach als Service installiert werden kann, muss wie folgt vorgegangen werden: Die Datei wrapper.conf im Verzeichnis <INSTALLDIR>/config/ muss wie folgt angepasst werden (4 Änderungen):

```
# Port which the native wrapper code will attempt to connect to
wrapper.port=1778
# Name of the service
wrapper.ntservice.name=MediData Second MPC
# Display name of the service
wrapper.ntservice.displayname=MediData Second MPC
# Description of the service
wrapper.ntservice.description=MediData Second MPC
```

## 5.3.3 SMB File Server Zugriff

Ab MPC V1.8.0 können die Sende-, Empfangs- und Errorverzeichnisse und ab MPC V1.9.0 die Archivverzeichnisse und die Verzeichnisse für Senden ohne Sende-Steuerdateien (Kap. 3.5) auf einem SMB file server angelegt werden, d.h. sog. Microsoft Windows "share" Verzeichnisse können hierfür verwendet werden. Im Verzeichnispfad muss nun das smb Protokoll angegeben werden, z.B.

```
CLIENT.1.SEND_DIR=smb://dataserver/testuser/data/send
```

**Wichtig:** Wenn das `smb://` Protokoll eingesetzt werden soll, müssen immer alle Verzeichnisse, welche für `smb://` konfigurierbar sind, dafür konfiguriert werden, nicht nur einige hiervon, d.h. pro Mandant *n* müssen immer alle 6 Properties `CLIENT.n.SEND_DIR`, `CLIENT.n.SENDARCHIVE_DIR`, `CLIENT.n.RECEIVE_DIR`, `CLIENT.n.RECEIVETEST_DIR`, `CLIENT.n.ERROR_DIR` sowie `CLIENT.n.DIR.m` mit `smb://` konfiguriert werden (falls sie verwendet werden). Dies sollte kein Problem sein, denn auch wenn sich einige dieser Verzeichnisse nur auf dem lokalen Gerät befinden, können diese trotzdem als Windows "share" Verzeichnisse freigegeben werden.

Ferner müssen entsprechende Properties gesetzt werden, deren Bezeichnung jeweils mit "jcifs" beginnt. Im einfachsten Fall muss nur Username, Passwort und evtl. Domain gesetzt werden:

### SMB file server access mpcommunicator.config (advanced)

Key	Bedeutung	Default
jcifs.smb.client.username	default Username	-
jcifs.smb.client.password	default Passwort	-
jcifs.smb.client.domain	default Domain	-
usw.	über 50 Properties können bei Bedarf gesetzt werden	-

Die Liste aller möglichen Properties, die gesetzt werden können, ist zu finden auf <http://jcifs.samba.org/src/docs/api/>

## 5.4 Konfiguration des Loggers

### 5.4.1 Allgemein

Anpassen des Logging Konfigurationsfiles <DATADIR>/config/mpcommunicator.logger

- Logfile (key: log4j.appender.A1.File, default: data/log/mpcommunicator.log)
- Maximale grösse einer Logdatei (key: log4j.appender.A1.MaxFileSize, default:500KB)
- Anzahl der Backuplogfiles bis diese wieder überschrieben werden (key: log4j.appender.A1.MaxBackupIndex, default:3)

### 5.4.2 Mail (SMTP)

Ab Version 1.3.x besteht die Möglichkeit im Fehlerfall sich eine Email Meldung zu senden. Dazu muss wie folgt vorgegangen werden:

Die zweite Zeile des Konfigurationsfiles muss von

```
log4j.rootCategory= INFO, A1, A2
```

nach

```
log4j.rootCategory= INFO, A1, A2, EMAIL
```

geändert werden. Zudem muss mindestens folgendes eingestellt werden:

- Mail Empfänger (key: log4j.appender.EMAIL.To)
- Mail Absender (key: log4j.appender.EMAIL.From)
- Mail / Smtip Server (key: log4j.appender.EMAIL.SMTPHost)

Falls das eMail an mehrere Empfänger gesendet werden soll, so müssen diese mit einem Komma („“,“) getrennt werden.

Es können so Log-Events der Kategorie ERROR oder FATAL per Email versandt werden.

Ab der MPC Version 1.20 wird per default der erste solche Fehler per Email versandt, danach wird höchstens eine weitere Email-Nachricht pro 10-Minuten-Intervall versandt. Dieser Default von 10 Minuten kann durch Verwenden der folgenden Property mit Angabe des Intervalles in Sekunden neu konfiguriert (Beispiel: 5 Minuten) werden

```
mpcommunicator.log4j.triggerdelay=300
```

und ist in der Datei

```
mpcommunicator.config
```

einzuführen.

Um sicherzustellen, dass dieses Zeitintervall beachtet wird, muss die folgende Property

```
log4j.appender.EMAIL.EvaluatorClass=com.medidata.mediport.communicator.utility.Log4jTriggerAfterDelay
```

in der Datei

```
mpcommunicator.logger
```

vorhanden sein. Wenn diese nicht vorhanden ist, wird wie in MPC Versionen <= 1.19 für jeden ERROR oder FATAL Log-Event direkt ein Email versandt.

### 5.4.3 SNMP Trap Sender

Die zweite Zeile des Konfigurationsfiles muss von

```
log4j.rootCategory=INFO, A1, A2 nach log4j.rootCategory=INFO, A1, A2, SNMP
```

geändert werden.

Wenn auf dem MPC Fehler auftreten, kann der MPC eine SNMP Nachricht erzeugen und diese über das Netzwerk dem SNMP Management Server zur Verfügung stellen. Die folgenden fett gedruckten Einstellungen (values) können konfiguriert werden.

SNMP Trap mpcommunicator.logger		
Key	Bedeutung	Default
log4j.appender.SNMP.ManagementHost	Adresse des Hosts zu dem die SNMP Traps gesendet werden sollen.	-
log4j.appender.SNMP.ManagementHostTrapListenPort	Port des Hosts zu dem die SNMP Traps gesendet werden sollen.	162
log4j.appender.SNMP.EnterpriseOID	EnterpriseOID	-

SNMP Trap mpcommunicator.logger		
log4j.appender.SNMP.LocalIPAddress	IP Adresse des Rechners auf dem der MPC installiert ist. Diese Information wird dann im Feld "agent"s IP address an den Host gesendet	-
log4j.appender.SNMP.LocalTrapSendPort	Port des Rechners von dem die SNMP Traps gesendet werden sollen.	163
log4j.appender.SNMP.GenericTrapType	Generischer Trap Typ (definiert durch SNMP Standard). Mögliche Werte sind: 0 -- cold start 1 -- warm start 2 -- link down 3 -- link up 4 -- authentication failure 5 -- EGP neighbor loss 6 -- enterprise specific	6
log4j.appender.SNMP.SpecificTrapType	Applikationsspezifischer Trap Tap. Wert zwischen -128 und 127.	0
log4j.appender.SNMP.ApplicationTrapOID	ApplicationTrapOID	-
log4j.appender.SNMP.CommunityString		public

## Beispiel:

```
log4j.appender.A3=org.apache.log4j.ext.SNMPTrapAppender
log4j.appender.A3.ImplementationClassName=org.apache.log4j.ext.JoeSNMPTrapSender
log4j.appender.A3.ManagementHost=192.168.1.99
log4j.appender.A3.ManagementHostTrapListenPort=162
log4j.appender.A3.EnterpriseOID=1.3.6.1.4.1.24.0
log4j.appender.A3.LocalIPAddress=192.168.1.99
log4j.appender.A3.LocalTrapSendPort=163
log4j.appender.A3.GenericTrapType=5
log4j.appender.A3.SpecificTrapType=0
log4j.appender.A3.ApplicationTrapOID=1.3.6.1.4.1.24.2
log4j.appender.A3.CommunityString=public
log4j.appender.A3.SysUpTime=1000
```

## 5.5 Test der Konfiguration

Nach der Konfiguration kann eine MPC Installation einfach getestet werden. Nachdem der MPC gestartet wurde, sollte im Log bald einmal der Eintrag "I2905: PARTNER changed[nnnn]" geschrieben werden, womit die erfolgreiche Verbindung zum MediPort Server erwiesen ist. Die übrigen Funktionen sollten dann höchstwahrscheinlich auch funktionieren und müssen nicht notwendigerweise noch getestet werden.

Ggf. können die einzelnen Funktionen manuell mittels RMI Betrieb (rmi.enable=true setzen) getestet werden. Um sowohl das Senden als auch das Empfangen zu testen, kann man ein

Test-Dokument "im Kreis herum" an sich selbst senden, indem man für die Absender- als auch für die Empfänger-GLN in beiden Fällen die eigene GLN einsetzt und das Dokument im Tiers Payant oder direct Workflow sendet (da Tiers Garant Dokumente zuerst eine Status-Änderung benötigen, bevor sie vom MPC empfangen werden können).

## 6. Update

### 6.1 Update

Da jeweils eine grössere Anzahl Dateien betroffen ist, sollte die alte Version des MPC deinstalliert werden (vgl. Kapitel 9) und die neue Version danach installiert werden. Hierbei ist es ggf. empfehlenswert, die alte(n) Konfigurationsdatei(en) vorübergehend anderswo zu sichern, damit diese bei der Neuinstallation nicht mit den Default-Werten überschrieben werden, damit nicht alle Einstellungen wieder ganz neu gemacht werden müssen. Durch einfaches Vergleichen der alten und neuen Konfigurationsdatei(en) kann so schnell ermittelt werden, welche Properties in der neuen Version hinzugekommen sind. I.d.R. ist nur die eine Datei `config/mpcommunicator.config` betroffen, und eventuell noch die Datei `config/mpcommunicator.logger`, falls in dieser auch individuelle Einstellungen gemacht wurden.

#### **Hinweis:**

Für die Deinstallation unter Max OS X bitte Kapitel 9.2 lesen.

## 7. Betrieb

### 7.1 Allgemein

#### 7.1.1 Starten

Der MediPort Communicator kann als RMI Server oder als Applikation mit eingebautem Scheduler gestartet werden. Es darf immer nur einer der beiden laufen. Im Gegensatz zum Scheduler können beim RMI Server die verschiedenen Tasks von aussen getriggert werden.

#### Scheduler

<INSTALLDIR>/MPC\_Scheduler\_Start

#### RMI Server

Damit der RMI Server gestartet werden kann, muss im Konfigurationsfile `rmi.enable=true` gesetzt werden.

<INSTALLDIR>/MPC\_RMI\_Start

#### 7.1.2 Arbeiten

- Im Verzeichnis <DATADIR>/data/receive resp. <DATADIR>/data/receive/test (falls es sich um Testdokumente handelt) werden vom Server geholte Dokumente abgelegt.
- Im Verzeichnis <DATADIR>/data/send können Sie zu sendende XML Dokumente ablegen und dann die dazugehörige Steuerdatei ablegen. **Achtung:** Die zugehörige XML-Schema Datei muss sich ebenfalls in diesem Verzeichnis befinden und darf von dort nicht gelöscht werden.
- Im Verzeichnis <DATADIR>/data/log befinden sich die erzeugten Logfiles.
- Im Verzeichnis <DATADIR>/data/partner befindet sich das erzeugte Partnerverzeichnis.
- Im Verzeichnis <DATADIR>/data/docstatus befinden sich die erzeugten Documentstati.

#### 7.1.3 Stoppen

ANMERKUNG: das Stoppen des MPCCommunicator kann bis zu etwa 20 Sekunden dauern.

Die einzelnen Task verhalten sich bei einem Stop wie folgt:

- Send: Schließt den Versand des aktuellen Dokumentes ab und speichert sich dann die Verarbeitungsposition innerhalb des SendControl Files um bei einem Neustart an dieser Stelle fortfahren zu können.
- Receive: Schließt den Empfang des aktuellen Dokumentes ab und beendet dann die Verbindung zum Server. Die auf dem Server verbleibenden Dokumente werden beim Neustart vom Server geholt.

- PartnerUpdate: Bricht das Erzeugen des Partnerverzeichnis Files ab und erzeugt ein neues Partnerverzeichnis File beim nächsten Neustart. Achtung: In diesem Falle kann es sein, dass das Partnerverzeichnis File kurzzeitig nicht mehr vorhanden ist (es wird jedoch zuvor nach PartnerInfo.txt\_OLD umkopiert) .
- GetDocstatus: Bricht das Erzeugen des Documentstati Files ab und erzeugt ein neues Documentstati File beim nächsten Neustart.

## **Scheduler**

<INSTALLDIR>/MPC\_Scheduler\_Stop

## **RMI Server**

<INSTALLDIR>/MPC\_RMI\_Stop

## 7.2 Windows spezifisch

### 7.2.1 Starten

#### Konsole (CMD)

- cd <INSTALLDIR>/MPCCommunicator/bin
- StartMPCCommunicator.bat

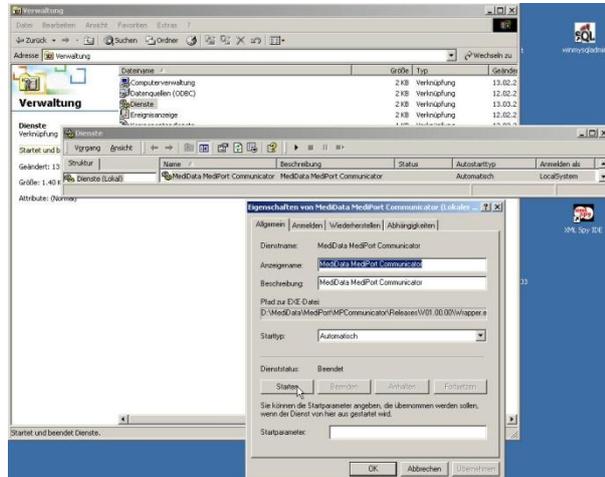
#### Explorer

Doppelklicken auf <INSTALLDIR>/MPCCommunicator/bin/ StartMPCCommunicator.bat

#### Windows Service

Nach der Installation startet der „MediData MediPort Communicator“ Service bei jedem Neustart des Computers automatisch mit dem Systemaccount( also auch ohne dass ein Benutzer sich am Computer eingeloggt hat).

Der Service kann jedoch auch über das Dienste Programm von Windows manuell gestartet werden:



### 7.2.2 Stoppen

#### Konsole und Explorer

Es laufen defaultmässig 2 Tasks (FileStop Task um die Applikation über ein Stopfile anzuhalten und ConsoleStop Task um die Applikation über die Konsole stoppen zu können) die für das Anhalten des MediPort Communicators zuständig sind:

- durch die Eingabe von "exit" und drücken der Return Taste in der gestarteten Konsole.
- durch erzeugen eines Files " mpcommunicator\_stop" im Verzeichnis <DATADIR>/data (Der Inhalt des Files spielt dabei keine Rolle)

## Windows Service

Über das Windows Dienste Programm analog dem Start

## 8. Unterhalt

### 8.1 Zertifikatswechsel

Einmal pro Jahr müssen die neuen SSL Zertifikate eingespielt werden. Kopieren Sie den Keystore von der Zertifikats-CD ins Installationsverzeichnis:

- Neues Keystorefile (wird separat geliefert) zur Verfügung stellen:
  - Kopieren des Keystorefiles EANxxxxxxxxxxxxx\_mpg.keystore in das Verzeichnis <DATADIR>/config/ (xxxxxxxxxxxxx steht für die EAN-Nummer Ihres Zertifikats)
  - MPC neu starten

#### Hinweis:

Massgebend für den Namen des vom MPC gelesenen Keystorefiles ist die Property keystore.name. Wenn der ab Version 1.9.0 defaultmässig gesetzte Name

```
keystore.name=config/EANxxxxxxxxxxxxx_mpg.keystore
```

unverändert beibehalten wird, muss das Keystorefile ab dieser Version nicht mehr wie früher in mpcommunicator.keystore umbenannt werden, sondern es wird ein Keystorefile mit der in der Property mediport.dn konfigurierten EAN-Nummer erwartet. Falls man jedoch wie früher keystore.name=config/mpcommunicator.keystore einsetzt, muss das Keystorefile wieder entsprechend umbenannt werden.

### 8.2 Applikation

- Aufräumen der Logfiles: Da zyklische Logfiles mit einer vorgegebenen Maximalgröße geschrieben werden, ist ein aufräumen der Logfiles nicht nötig).
- Überwachen / Löschen der Files im Error Verzeichnis
- Überwachen / Löschen der Files im Receive Verzeichnis
- Überwachen / Löschen der Files im Docstatus Verzeichnis

### 8.3 Datenbank

Zum Unterhalt der Datenbank sind keine speziellen Schritte nötig. Da sich in der Datenbank außer dem Partnerverzeichnis keine dynamischen Daten befinden und das Partnerverzeichnis bei einer eventuellen Neuinstallation automatisch aktualisiert wird, ist kein spezielles Backup der Datenbank nötig.

## 9. Deinstallation

### 9.1 Allgemein

- Falls der MP Communicator als Windows Service installiert wurde, muss <INSTALLDIR>/bin/UninstallMPCommunicatorServiceWrapper-Win.bat ausgeführt werden.
- Starten des Uninstallers in <INSTALLDIR>/UninstallerData/.

#### Hinweis:

Es werden nur die vom Installer installierten Dateien auf der Harddisk entfernt. Eventuell später hinzugekommene Dateien (z.B. Logdateien, empfangene Dokumente, ...) werden bei der Deinstallation NICHT gelöscht und müssen manuell entfernt werden.

### 9.2 Deinstallation unter Mac OS X

Unter Mac OS X Betriebssystem können Programme einfach deinstalliert werden, indem alle installierten Dateien im Papierkorb gelegt werden.

Legen Sie den <INSTALLDIR>/MPCommunicator Ordner im Papierkorb.

## 10. Problembehandlung

### 10.1 Problem mit Microsoft Proxy

Es wird nur BASIC authentication unterstützt (keine NTLM authentication).

recently we also came across some problems with authentication towards a Microsoft proxy. The reason was that the proxy was configured to do NTLM authentication and not BASIC authentication. Unfortunately, NTLM authentication is only supported by Microsoft components such as Internet Explorer (the NTLM protocol is proprietary to Microsoft :-). The only way to get around this is probably to install the Microsoft WinSock Proxy Client which handles NTLM authentication transparently to any application.

## 11. Anhang A: Dateinamen der empfangenen Dokumente

**Dateiname XML:**    bbbbbbbbtntttttttttttaiin.XML

### Codierung Dateinamen:

bbbbbbbbbb            Schlüssel gemäss Bezeichnung in Tabelle unten  
 tttttttttttttt        Dokument erstmals gespeichert am <Zeit/Datum> in 1/1000  
                               Sekunden seit 1.1.1970  
 a                         Art des Dokuments T = Test; P = Produktiv  
 ii                        Kennung des Intermediärs, der das Dokument transportierte.  
 n                        Laufnummer für Dokumente in gleicher Millisekunde  
 XML                     Dateiendung für xml-Dokumente

<b>Auflösung Schlüssel des Dokumentennamens (Teil bbbbbbbbb)</b>		
<b>Dokument</b>	<b>Schema</b>	<b>Dateiname</b>
Zahnarzt-Rechnung, Tiers Payant	DentistInvoiceRequest_300.xsd	<b>IRDEP0300</b>
Zahnarzt-Rechnung, Tiers Garant	DentistInvoiceRequest_300.xsd	<b>IRDEM0300</b>
Antwort auf eine Zahnarzt-Rechnung	DentistInvoiceResponse_300.xsd	<b>ISDEX0300</b>
Spital-Rechnung, Tiers Payant	HospitalInvoiceRequest_300.xsd	<b>IRHSP0300</b>
Spital-Rechnung, Tiers Garant	HospitalInvoiceRequest_300.xsd	<b>IRHSM0300</b>
Antwort auf eine Spital-Rechnung	HospitalInvoiceResponse_300.xsd	<b>ISHSX0300</b>
Rechnungs-Mahnung	InvoiceReminderRequest_300.xsd	<b>IRREX0300</b>
Antwort auf eine Rechnungs-Mahnung	InvoiceReminderResponse_300.xsd	<b>ISREX0300</b>
Labor-Rechnung, Tiers Payant	LabInvoiceRequest_300.xsd	<b>IRLAP0300</b>
InvoiceInspector geprüfte Labor-Rechnung, Tiers Payant	QO_LabInvoiceRequest_300.xsd	<b>QRLAP0300</b>
Labor-Rechnung, Tiers Garant	LabInvoiceRequest_300.xsd	<b>IRLAM0300</b>
InvoiceInspector geprüfte Labor-Rechnung, Tiers Garant	QO_LabInvoiceRequest_300.xsd	<b>QRLAM0300</b>
Antwort auf eine Labor-Rechnung	LabInvoiceResponse_300.xsd	<b>ISLAX0300</b>
Arzt-Rechnung, Tiers Payant	MDInvoiceRequest_300.xsd	<b>IRMDP0300</b>
InvoiceInspector geprüfte Arzt-Rechnung, Tiers Payant	QO_MDInvoiceRequest_300.xsd	<b>QRMDP0300</b>
Arzt-Rechnung, Tiers Garant	MDInvoiceRequest_300.xsd	<b>IRMDM0300</b>
InvoiceInspector geprüfte Arzt-Rechnung, Tiers Garant	QO_MDInvoiceRequest_300.xsd	<b>QRMDM0300</b>
Antwort auf eine Arzt-Rechnung	MDInvoiceResponse_300.xsd	<b>ISMDX0300</b>
Apotheker-Rechnung, Tiers Payant	PharmacyInvoiceRequest_300.xsd	<b>IRPAP0300</b>
Apotheker-Rechnung, Tiers Garant	PharmacyInvoiceRequest_300.xsd	<b>IRPAM0300</b>

<b>Auflösung Schlüssel des Dokumentennamens (Teil bbbbbbbbbb)</b>		
<b>Dokument</b>	<b>Schema</b>	<b>Dateiname</b>
Antwort auf eine Apotheker-Rechnung	PharmacyInvoiceResponse_300.xsd	<b>ISPAX0300</b>
Physiotherapeuten-Rechnung, Tiers Payant	PhysiolInvoiceRequest_300.xsd	<b>IRPYP0300</b>
Physiotherapeuten-Rechnung, Tiers Garant	PhysiolInvoiceRequest_300.xsd	<b>IRPYM0300</b>
Antwort auf eine Physiotherapeuten-Rechnung	PhysiolInvoiceResponse_300.xsd	<b>ISPYX0300</b>
Schuhmacher-Rechnung, Tiers Payant	ShoemakerInvoiceRequest_300.xsd	<b>IRSMP0300</b>
Schuhmacher-Rechnung, Tiers Garant	ShoemakerInvoiceRequest_300.xsd	<b>IRSMM0300</b>
Antwort auf eine Schuhmacher-Rechnung	ShoemakerInvoiceResponse_300.xsd	<b>ISSMX0300</b>
Rechnung "Übrige", Tiers Payant	UnspecifiedInvoiceRequest_300.xsd	<b>IRUNP0300</b>
Rechnung "Übrige", Tiers Garant	UnspecifiedInvoiceRequest_300.xsd	<b>IRUNM0300</b>
Antwort auf eine Rechnung "Übrige"	UnspecifiedInvoiceResponse_300.xsd	<b>ISUNX0300</b>
"MiGeL"-Rechnung, Tiers Payant	WholesalerInvoiceRequest_300.xsd	<b>IRWSP0300</b>
"MiGeL"-Rechnung, Tiers Garant	WholesalerInvoiceRequest_300.xsd	<b>IRWSM0300</b>
Antwort auf eine "MiGeL"-Rechnung	WholesalerInvoiceResponse_300.xsd	<b>ISWSX0300</b>
Unbekannter Dokumenttyp Tiers Payant	<keine Bezeichnung>	<b>UKDTP0000</b>
Unbekannter Dokumenttyp Tiers Garant	<keine Bezeichnung>	<b>UKDTM0000</b>
Zahnarztrechnung 4.0, Tiers Payant	DentistInvoiceRequest_400.xsd	<b>IRDEP0400</b>
Zahnarztrechnung 4.0, Tiers Garant	DentistInvoiceRequest_400.xsd	<b>IRDEM0400</b>
Antwort auf eine Zahnarztrechnung 4.0	DentistInvoiceResponse_400.xsd	<b>ISDEX0400</b>
TG Document Request 4.0	DocumentRequest_400.xsd	<b>TGDRX0400</b>
TG Document Response 4.0	DocumentResponse_400.xsd	<b>TGDSX0400</b>
Kostengutsprache 4.0	generalcreditrequest_400.xsd	<b>CRGEX0400</b>
Antwort auf eine Kostengutsprache 4.0	generalcreditresponse_400.xsd	<b>CSGEX0400</b>
Generelle Rechnung 4.1, Tiers Payant	generalinvoicerequest_410.xsd	<b>IRGEP0410</b>
InvoiceInspector geprüfte Generelle Rechnung 4.1, Tiers Payant	QO_generalinvoicerequest_410.xsd	<b>QRGEP0410</b>
Generelle Rechnung 4.1, Tiers Garant	generalinvoicerequest_410.xsd	<b>IRGEM0410</b>
InvoiceInspector geprüfte Generelle Rechnung 4.1, Tiers Garant	QO_generalinvoicerequest_410.xsd	<b>QRGEM0410</b>
Antwort auf eine Generelle Rechnung 4.1	generalinvoiceresponse_410.xsd	<b>ISGEX0410</b>
Spital Kostengutsprache 4.0	hospitalcreditrequest_400.xsd	<b>CRHSX0400</b>
Antwort auf eine Spital Kostengutsprache 4.0	hospitalcreditresponse_400.xsd	<b>CSHSX0400</b>
Spital-Rechnung 4.0, Tiers Payant	HospitalInvoiceRequest_400.xsd	<b>IRHSP0400</b>
InvoiceInspector geprüfte Spital-Rechnung 4.0, Tiers Payant	QO_HospitalInvoiceRequest_400.xsd	<b>QRHSP0400</b>
Spital-Rechnung 4.0, Tiers Garant	HospitalInvoiceRequest_400.xsd	<b>IRHSM0400</b>
InvoiceInspector geprüfte Spital-Rechnung 4.0, Tiers Garant	QO_HospitalInvoiceRequest_400.xsd	<b>QRHSM0400</b>
Antwort auf eine Spital-Rechnung 4.0	HospitalInvoiceResponse_400.xsd	<b>ISHSX0400</b>
Rechnungs-Mahnung 4.0	InvoiceReminderRequest_400.xsd	<b>IRREX0400</b>
Antwort auf eine Rechnungs-Mahnung 4.0	InvoiceReminderResponse_400.xsd	<b>ISREX0400</b>

<b>Auflösung Schlüssel des Dokumentennamens (Teil bbbbbbbbbb)</b>		
<b>Dokument</b>	<b>Schema</b>	<b>Dateiname</b>
Laborrechnung 4.0, Tiers Payant	LabInvoiceRequest_400.xsd	<b>IRLAP0400</b>
Laborrechnung 4.0, Tiers Garant	LabInvoiceRequest_400.xsd	<b>IRLAM0400</b>
Antwort auf eine Laborrechnung 4.0	LabInvoiceResponse_400.xsd	<b>ISLAX0400</b>
Arzt-Rechnung 4.0, Tiers Payant	MDInvoiceRequest_400.xsd	<b>IRMDP0400</b>
InvoiceInspector geprüfte Arzt-Rechnung 4.0, Tiers Payant	QO_MDInvoiceRequest_400.xsd	<b>QRMDP0400</b>
Arzt-Rechnung 4.0, Tiers Garant	MDInvoiceRequest_400.xsd	<b>IRMDM0400</b>
InvoiceInspector geprüfte Arzt-Rechnung 4.0, Tiers Garant	QO_MDInvoiceRequest_400.xsd	<b>QRMDM0400</b>
Antwort auf eine Arzt-Rechnung 4.0	MDInvoiceResponse_400.xsd	<b>ISMDX0400</b>
Apotheker-Rechnung 4.0, Tiers Payant	PharmacyInvoiceRequest_400.xsd	<b>IRPAP0400</b>
InvoiceInspector geprüfte Apotheker-Rechnung 4.0, Tiers Payant	QO_PharmacyInvoiceRequest_400.xsd	<b>QRPAP0400</b>
Apotheker-Rechnung 4.0, Tiers Garant	PharmacyInvoiceRequest_400.xsd	<b>IRPAM0400</b>
InvoiceInspector geprüfte Apotheker-Rechnung 4.0, Tiers Garant	QO_PharmacyInvoiceRequest_400.xsd	<b>QRPAM0400</b>
Antwort auf eine Apotheker-Rechnung 4.0	PharmacyInvoiceResponse_400.xsd	<b>ISPAX0400</b>
Physiotherapeutenrechnung 4.0, Tiers Payant	PhysiolInvoiceRequest_400.xsd	<b>IRPYP0400</b>
Physiotherapeutenrechnung 4.0, Tiers Garant	PhysiolInvoiceRequest_400.xsd	<b>IRPYM0400</b>
Antwort auf Physiotherapeutenrechnung 4.0	PhysiolInvoiceResponse_400.xsd	<b>ISPYX0400</b>
Schuhmacherrechnung 4.0, Tiers Payant	ShoemakerInvoiceRequest_400.xsd	<b>IRSMP0400</b>
Schuhmacherrechnung 4.0, Tiers Garant	ShoemakerInvoiceRequest_400.xsd	<b>IRSM0400</b>
Antwort auf eine Schuhmacherrechnung 4.0	ShoemakerInvoiceResponse_400.xsd	<b>ISSMX0400</b>
Nicht definierte Rechnung 4.0, Tiers Payant	UnspecifiedInvoiceRequest_400.xsd	<b>IRUNP0400</b>
Nicht definierte Rechnung 4.0, Tiers Garant	UnspecifiedInvoiceRequest_400.xsd	<b>IRUNM0400</b>
Antwort auf eine Nicht definierte Rechnung 4.0	UnspecifiedInvoiceResponse_400.xsd	<b>ISUNX0400</b>
Migelrechnung 4.0, Tiers Payant	WholesalerInvoiceRequest_400.xsd	<b>IRWSP0400</b>
Migelrechnung 4.0, Tiers Garant	WholesalerInvoiceRequest_400.xsd	<b>IRWSM0400</b>
Antwort auf eine Migelrechnung 4.0	WholesalerInvoiceResponse_400.xsd	<b>ISWSX0400</b>
xmit Dokumente 4.0	xmitdocuments_400.xsd	<b>XMGEX0400</b>
xmit Dokumente 4.1	xmitDocuments_410.xsd	<b>XMGEX0410</b>
Genereller Container 4.3, Invoice Tiers Payant	generalContainer_430.xsd	<b>GCUKP0430</b>
InvoiceInspector geprüfter Genereller Container 4.3, Invoice Tiers Payant	QO_generalContainer_430.xsd	<b>QCUKP0430</b>
Genereller Container 4.3, Invoice Tiers Garant	generalContainer_430.xsd	<b>GCUKM0430</b>
InvoiceInspector geprüfter Genereller Container 4.3, Invoice Tiers Garant	QO_generalContainer_430.xsd	<b>QCUKM0430</b>
Genereller Container 4.3, nur MCD	generalContainer_430.xsd	<b>GCUKX0430</b>
Generelle Kostengutsprache Anfrage 4.3	generalCreditRequest_430.xsd	<b>CRGEX0430</b>
Generelle Kostengutsprache Antwort 4.3	generalCreditResponse_430.xsd	<b>CSGEX0430</b>

<b>Auflösung Schlüssel des Dokumentennamens (Teil bbbbbbbbbb)</b>		
<b>Dokument</b>	<b>Schema</b>	<b>Dateiname</b>
Formular Transport Anfrage 4.3	generalFormRequest_430.xsd	<b>FRGEX0430</b>
Formular Transport Antwort 4.3	generalFormResponse_430.xsd	<b>FSGEX0430</b>
Generelle Rechnung 4.3, Tiers Payant	generalInvoiceRequest_430.xsd	<b>IRGEP0430</b>
InvoiceInspector geprüfte Generelle Rechnung 4.3, Tiers Payant	QO_generalInvoiceRequest_430.xsd	<b>QRGEP0430</b>
Generelle Rechnung 4.3, Tiers Garant	generalInvoiceRequest_430.xsd	<b>IRGEM0430</b>
InvoiceInspector geprüfte Generelle Rechnung 4.3, Tiers Garant	QO_generalInvoiceRequest_430.xsd	<b>QRGEM0430</b>
Generelle Rechnung 4.4, Tiers Payant	generalInvoiceRequest_440.xsd	<b>IRGEP0440</b>
InvoiceInspector geprüfte Generelle Rechnung 4.4, Tiers Payant	QO_generalInvoiceRequest_440.xsd	<b>QRGEP0440</b>
Generelle Rechnung 4.4, Tiers Garant	generalInvoiceRequest_440.xsd	<b>IRGEM0440</b>
InvoiceInspector geprüfte Generelle Rechnung 4.4, Tiers Garant	QO_generalInvoiceRequest_440.xsd	<b>QRGEM0440</b>
Antwort auf Generelle Rechnung 4.3	generalInvoiceResponse_430.xsd	<b>ISGEX0430</b>
Antwort auf Generelle Rechnung 4.4	generalInvoiceResponse_440.xsd	<b>ISGEX0440</b>
Mitteilung 4.3	generalNotification_430.xsd	<b>NFGEX0430</b>
Mitteilung 4.4	generalNotification_440.xsd	<b>NFGEX0440</b>
Anfrage Spital MCD 4.3	hospitalMCDRequest_430.xsd	<b>MRHSX0430</b>
Anfrage Spital MCD 4.4	hospitalMCDRequest_440.xsd	<b>MRHSX0440</b>
Antwort Spital MCD 4.3	hospitalMCDResponse_430.xsd	<b>MSHSX0430</b>
Antwort Spital MCD 4.4	hospitalMCDResponse_440.xsd	<b>MSHSX0440</b>
Anfrage Status 4.3	statusRequest_430.xsd	<b>SRUKX0430</b>
Antwort Status 4.3	statusResponse_430.xsd	<b>SSUKX0430</b>
EBPP Rechnungsstellungsnachricht	EbppInvoiceRequest_100.xsd	<b>IREBX0100</b>
EBPP Administrative Nachricht	EbppAdminMessage_100.xsd	<b>AMEBX0100</b>
eKARUS Generelle Rechnung, Tiers Payant	ekarus_x230.xsd	<b>EKHSP0230</b>
eKARUS Generelle Rechnung, Tiers Garant	ekarus_x230.xsd	<b>EKHSM0230</b>
eKARUS Anfrage Spital MCD	ekarus_x240.xsd	<b>EKHSX0240</b>
eKARUS Antwort auf Generelle Rechnung	ekarus_x250.xsd	<b>EKHSX0250</b>

Informationen über aktuelle XML-Schemas können unter [www.forum-datenaustausch.ch](http://www.forum-datenaustausch.ch) bezogen werden.

## Auflösung des Dateinamensprefix:

### IRDEP0300 1-2: Dokumenttyp

IR = InvoiceRequest	(Rechnung)
IS = InvoiceReSponse	(Rechnungs-Antwort)
CR = CreditRequest	(Kostengutsprache)
CS = CreditReSponse	(Kostengutsprache-Antwort)
XM = XmitDocuments	(xmit Dokumente)
MR = McdRequest	(Minimal clinical dataset)
MS = McdReSponse	(Minimal clinical dataset-Antwort)
SR = StatusRequest	(Status)
SS = StatusReSponse	(Status-Antwort)
GC = GeneralContainer	(Rechnung und/oder MCD)
FR = medForm Request	(Formular Transport Anfrage)
FS = medForm ReSponse	(Formular Transport Antwort)
NF = Notification	(Mitteilung)
EK = eKARUS	(eKARUS)
QR = QO-InvoiceRequest	(InvoiceInspector geprüfte Rechnung)
QC = QO-GeneralContainer	(InvoiceInspector geprüfter Container)

### IRDEP0300 3-4: Rolle

<b>DE = DE</b> n <del>ti</del> st	(Zahnarzt)
<b>HS = HoS</b> pital	(Spital)
<b>RE = RE</b> minder	(Mahnung)
<b>LA = LA</b> b	(Labor)
<b>MD = MedicalD</b> octor	(Arzt)
<b>PA = PhA</b> r <del>ma</del> cy	(Apotheke)
<b>PY = PhY</b> sio	(Physiotherapie)
<b>SM = Sho</b> eM <del>ak</del> er	(Schuhmacher/Orthopädietechniker)
<b>UN = UN</b> specified	(Übrige)
<b>ED = UN/ED</b> IFACT	(Edifact)
<b>WS = W</b> holeS <del>al</del> er	(MiGeL)
<b>UK = UnK</b> nown	(Unbekannter Typ)
<b>GE = GE</b> neral	(generell)
<b>EB = EB</b> PP (EBPP)	

### IRDEP0300 5: Vergütungsart

<b>P = Tiers P</b> ayant
<b>M = Tiers G</b> arant
<b>X = Nicht relevant</b>

IRDEP**0300** 6-9: Release

**0300** = 3.00

**0400** = 4.00

**0410** = 4.10

**0430** = 4.30

**0440** = 4.40

**0230** = eKARUS InvoiceRequest

**0240** = eKARUS MCDRequest

**0250** = eKARUS InvoiceResponse

**0000** = nicht relevant

## 12. Anhang B: Log Meldungen (Auszug)

### 12.1 Meldungen vom Log Level INFO (I0000 – I9999)

Info Nr.	Grund	Ausgabe
I0801	Ein neues Partnerverzeichnis wurde erstellt. Das alte Partnerverzeichnis wurde nach ..._OLD umkopiert.	New partnerfile[<filename>] created.
I1001	Es wird mit der Verarbeitung eines neuen SendControlFiles begonnen.	Start processing new SendControlFile[<sendcontrolfileinfo>]
I2201	Startup Meldung mit Versionsangabe	***** *** Start MediPort Communicator *** to stop: *** exit<ret> or Put mpcommunicator_stop file *** into data directory ***** Starting MPCommunicator V1.0.0
I2203	Shutdown Meldung mit Statistik	RunningSince[<date>] NbrOfJobsExecuted[<nbr>] NbrDocs Sent[<nbr>] NotSent[<nbr>] Received[<nbr>]
I2204	Scheduler bereitet Tasks vor	"----- Scheduler schedule jobs -----"
I2205	Info über jeden auszuführenden Task	grp_mg.<taskname> will run at[<date_of_first_run>] & repeat[<repeat_count>/<repeat_interval_in_m s>] or grp_mg.<taskname> will run at[<date_of_first_run>] with cronExpression[<cron_expression>]
I2206	Scheduler wurde gestartet	"----- Scheduler started -----"
I2207	Scheduler hat die Stopmeldung bekommen und wartet nun darauf, dass alle laufenden Tasks sich beenden.	"----- Shutting Down -----"
I2208	Scheduler wurde angehalten. Alle Tasks sind gestoppt.	"----- Shutdown Complete -----"
I2401	Ein Dokument wurde erfolgreich empfangen.	RECEIVE DOC OK FILENAME[<filename>] SENDERDN[<senderdn>] STATEID[200] MDID[<doc_id>] DOCTYPE[<doc_typ>] JOB[<job_id>]
I2501	Ein Dokument wurde erfolgreich gesendet.	SEND DOC OK FILENAME[<filename>] RECEIVERDN[<receiver_dn>] STATEID[100] MDID[<doc_id>] DOCTYPE[<doc_typ>] JOB[<job_id>]
I2902	Der PartnerUpdate Task hat festgestellt, dass das Partnerverzeichnis auf dem Server sich nicht geändert hat.	ADRBOOK UNCHANGED
I2903	Der Receive Task hat festgestellt, dass sich keine neuen Dokumente auf dem Server befinden.	NO NEW DOCUMENTS ON SERVER
I2905	Der PartnerUpdate Task hat festgestellt, dass das Partnerverzeichnis auf dem Server <nbr> neue/geänderte Einträge enthält. Das neue Partnerverzeichnis auf der lokalen Harddisk ist jedoch zu diesem Zeitpunkt noch nicht nachgeführt!	PARTNER changed[<nbr>]
I2502	Der SendControlFile Task hat festgestellt, dass das Partnerverzeichnis nicht fertig erstellt wurde und dass dadurch noch nicht alle Partner vorhanden sind.	Aktueller Zeitstempel für das Partnerverzeichnis wurde nicht gefunden. Es wird angenommen, dass das Partnerverzeichnis nicht aktuell ist - Sendvorgang wird übersprungen.

## 12.2 Meldungen vom Log-Level WARN (W0000 – W9999)

Warn Nr.	Ausgabe	Grund	Folge
W0601	Property file not found. Create a new one.	Logging Konfigurationsfile nicht gefunden.	Es wird ein neues Logging Konfigurationsfile mit Defaultwerten erstellt.
W0801	< SQLException>	Fehler beim Schliessen von Connection/Stmt/ResultSet/File	
W0802	< SQLException>	Fehler beim Schliessen von Connection/Stmt/ResultSet/File	
W0803	< SQLException>	Fehler beim Schliessen von Connection/Stmt/ResultSet/File	
W0804	SQLWarning: 0 rows INSERTED into table SENDFILEINFO	Fehler beim Abspeichern der SendControlFile Position in der DB.	Beim nächsten Aufstarten des MPCcommunicator gibt es ein paar Fehlermeldungen, weil bereits abgehandelte Dokumente des SendControlFiles nicht mehr da sind.
W0805	< SQLException>	Fehler beim Schliessen von Connection/Stmt/ResultSet/File	
W0806	< SQLException>	Fehler beim Schliessen von Connection/Stmt/ResultSet/File	
W0807	SQLWarning: 0 rows Deleted from table SENDFILEINFO	Fehler beim Löschen der SendControlFile Einträge in der Datenbank.	Warnungen beim nächsten Start, da die Infos des SendControlFile in der Datenbank nicht mehr auf der Harddisk vorhanden sind. Falls bereits ein neues SendControlFile sich auf der Harddisk befindet, stimmt die filesize und/oder der lastmodified timestamp nicht mehr und es gibt deswegen eine Warnung.
W0808	< SQLException>	Fehler beim Schliessen von Connection/Stmt/ResultSet/File	
W0809	< SQLException>	Fehler beim Schliessen von Connection/Stmt/ResultSet/File	
W0810	< SQLException>	Fehler beim Schliessen von Connection/Stmt/ResultSet/File	
W0811		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0812		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0813		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0814		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0815		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0816		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0817		der neue Partner wurde nicht in die Datenbank eingefügt	da es keine FK und keine anderen Constraints auf der DB hat, kann dies eigentlich nicht passieren.
W0818		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0819		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0820		Sleep Thread wurde unterbrochen	
W0821		SQLException beim schliessen der Connection/Stmt/ResultSet	

Warn Nr.	Ausgabe	Grund	Folge
W0822		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0823		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0824		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0825		SQLException beim schliessen der Connection/Stmt/ResultSet	
W0826		abkürzung für schemaurl nicht in DB gefunden. Schemaname ev. nicht in DB eingetragen	benutzen von UNKN als FileIdentifier
W0827		SQLException beim Schliessen der Connection/Stmt/ResultSet	
W0901		Der DB Verbindungspool wurde erhöht.	
W1010		Der gesuchte InputChannel ist kein gültiger InputChannel	
W1101		Der gesuchte OutputCahnnel ist kein gültiger Outputchannel	
W1102		Exception beim schliessen vomInput / Outputstream.	
W1701		Methode nicht impl.	
W2001		Sleep Thread wurde unterbrochen	
W2101		stopfile konnte nicht gelöscht werden	wenn das stopfile noch da ist, dann führt ein erneutes starten der applikation sofort wieder zur beendigung.
W2201		SchedulerException during statistic output	
W2202		scheduler schon gestartet	
W2203		scheduler schon gestopt	
W2204		Sleep Thread wurde unterbrochen	
W2401		IOException:fehler beim schliessen der in(document) oder out(file) streams	
W2501		SQLException: Der Server hat ein Stopsignal erhalten und bricht nun die verarbeitung des SendControlFiles ab und versucht nun die Position im File in die DB zu schreiben, was momentan nicht geht.	Das MiniGateway versucht beim nächsten Start das SendControlFile wieder ganz von anfang an abzuarbeiten, findet die schon verarbeiteten dokumente nicht und protokolliert darum ein paar fehler. Danach werden die restlichen dokumente ordnungsgemäss abgearbeitet.
W2502		SQLException: Der Server kann das SendControlFile nicht aus der DB lesen.	siehe W0601
W2601		Einer der Jobs hat eine Exception geworfen und diese wird hier zentral abgefangen.	
W2701		Sleep Thread wurde unterbrochen	
W2901		unimplemented method HttpClient.isAvailable returns always false.	
W2902		Exception: das Sliessen der JDConnection warf eine exception.	
W2903		IOException: das schliessen des inputstreams ging nicht	
W2904		IOException: das schliessen des inputstreams ging nicht	
W2905		IOException: das schliessen des inputstreams ging nicht	

Warn Nr.	Ausgabe	Grund	Folge
W2906		IOException: das schliessen des inputstreams ging nicht	
W2907		IOException: das schliessen des inputstreams ging nicht	
W2908		IOException: das schliessen des inputstreams ging nicht	
W2909		IOException: das schliessen des inputstreams ging nicht	
W3601		Die in der DB gespeicherte CSendFileInfo enthält eine andere Filesize als das aktuelle SendControlFile auf der HD	Die Abarbeitung der aktuellen SendControlFiles beginnt ganz am anfang und überspringt keine dokumente.
W3602		Die in der DB gespeicherte CSendFileInfo enthält ein anderes LastModificationDate als das aktuelle SendControlFile auf der HD	Die Abarbeitung der aktuellen SendControlFiles beginnt ganz am Anfang und überspringt keine Dokumente.
W3603		Die in der DB gespeicherte CSendFileInfo enthält eine andere anzahl dokumente als das aktuelle SendControlFile auf der HD	Die Abarbeitung der aktuellen SendControlFiles beginnt ganz am Anfang und überspringt keine Dokumente.
W3604		In der DB sind infos über ein SendControlFile gespeichert, aber auf der HD befindet sich momentan keines.	Es werden momentan keine dokumente versendet.
W4201		laik provider ist schon registriert	
W4202		Das Certifikat auf dem Server wird demnächst gewechselt.	Es kann während dieser Zeit nicht mit dem Server kommuniziert werden.
W4203		IOException: der outputstream konnte nicht geschlossen werden.	
W5801	XML VALIDATION problem[<problem>]	Problem der XML Validierung	Je nach eingestelltem xml.validation.abort.level wird das zu versendende Dokument verworfen oder nicht.

## 12.3 Meldungen vom Log-Level ERROR (E0000 – E9999)

Error Nr.	Meldung	Grund	Folge	Action
E0801	outputAllPartner failed to move file <temp> to <new>	Umbenennen des neuen Temp PartnerInfo Files zum echten Partner Info File ging nicht	Das Partnerverzeichnis ist nicht mehr aktuell. Beim nächsten Start des PartnerUpdate Tasks wird erneut versucht ein neues Partnerverzeichnis zu erstellen	
E0802	outputAllPartner failed to move file <partnerfile> to <old>	Umbenennen des alten PartnerInfo files nach Old PartnerInfoFile ging nicht.	Das Partnerverzeichnis ist nicht mehr aktuell. Beim nächsten Start des PartnerUpdate Tasks wird erneut versucht ein neues Partnerverzeichnis zu erstellen	
E0803	Close database connection failed.	Schliessen der Datenbankverbindung nicht gelungen.	Ev. Gehen nach einer gewissen Zeit die Datenbankverbindungen aus und die Applikation muss deshalb neu gestartet werden.	
E1001	Can't read SendControlFile[<filename>]	Keine Leserechte auf SendControlFile	Die momentane Verarbeitung des SendControlFiles wird abgebrochen	Zugriffsrechte anpassen
E1002	Fehler im SendControlFile	NumberFormatException	Das SendControlFile wird nicht abgearbeitet aber dennoch wird versucht, es zu löschen.	
E1003	Fehler im SendControlFile	XmlParserException	Das SendControlFile wird nicht abgearbeitet aber dennoch wird versucht, es zu löschen.	

Error Nr.	Meldung	Grund	Folge	Action
E1004	Aus das SendControlFile kann nicht richtig zugegriffen werden.	IOException	Das SendControlFile wird nicht abgearbeitet aber dennoch wird versucht, es zu löschen.	Kontrolle der Zugriffsrechte.
E1005	File[<filename>] with URL[<url>] not found	Das im SendControlFile angegebene File existiert nicht.	Dieser Eintrag im SendControlFile wird übersprungen.	Manuell kontrollieren, wo dieses file sich befindet.
E1006	File[<filename>] has not the correct accessrights(rw)	Die benötigten Zugriffsrechte auf das zu verarbeitende File sind nicht vorhanden.	Dieser Eintrag im SendControlFile wird übersprungen.	Manuell kontrollieren, wo dieses file sich befindet und welche zugriffsrechte es hat..
E1007	File[<filename>] has size[<filesize>] but should have size[<filesize>]	Das File hat nicht die in der SendControlDatei angegebene Filesize.	Dieser Eintrag im SendControlFile wird übersprungen. Das File wird in das Error Verzeichnis verschoben.	Manuelle Kontrolle, wieso die Filegröße nicht stimmt.
E1008	Building a valid URL failed.	Zugriff auf das File war nicht möglich	Dieser Eintrag im SendControlFile wird übersprungen. Das File wird's ins Error Verzeichnis kopiert.	
E1009	Reading the inputfile failed.	Einlesen des Files war nicht möglich	Dieser Eintrag im SendControlFile wird übersprungen. Das File wird in das Error Verzeichnis verschoben.	
E1010	Closing of io-streams failed.	Abschliessen des Einlesens des Files war nicht möglich.	Dieser Eintrag im SendControlFile wird übersprungen. Das File wird in das Error Verzeichnis verschoben.	
E1101	Document can not be saved.	Die temporäre Datei(Endung TMP) in der das empfangenen Dokument abgespeichert wurde, kann nicht in das endgültige Dokument umbenannt werden.	Das Dokument gilt als nicht empfangen und wird beim nächsten Verbindungsaufbau zum Server von dort nochmals geholt. Die temporäre Datei bleibt jedoch im Eingangsverzeichnis.	Manuelles Löschen der temporären Datei (mit Endung TMP)
E1102	Targetfile cannot be generated.	Das vom Server empfangenen Dokument kann nicht auf die Harddisk geschrieben werden.	Das Dokument gilt als nicht empfangen und wird beim nächsten Verbindungsaufbau zum Server von dort nochmals geholt.	
E1103	IO Error during storing information.	Das vom Server empfangenen Dokument kann nicht auf die Harddisk geschrieben werden.	Das Dokument gilt als nicht empfangen und wird beim nächsten Verbindungsaufbau zum Server von dort nochmals geholt.	
E1201	Document [<docname>] cannot be deleted.	Löschen des zu sendenden Dokumentes nicht möglich	Das Dokument wird nicht gelöscht und bleibt im Eingangsverzeichnis stehen, dies bedeutet jedoch nicht zwingend, dass das Dokument nicht an den Server gesendet wurde. Um dies festzustellen, müssen die vorangehenden Meldungen analysiert werden.	Manuelles löschen dieses Files
E1202	move to errorchannel of document [<docname>] failed.	Verschieben des dokumentes in den ErrorChannel misslungen.	E2503	
E1801	Parsing of XML-content failed.	Fehler beim Parsen des empfangenen Dokumentes.	Empfang des momentan verarbeiteten Dokumentes vom Server wird abgebrochen und das nächste Dokument wird vom Server geholt. Das fehlerhafte Dokument wird beim nächsten Aufruf des Receive Tasks erneut vom Server geholt.	

Error Nr.	Meldung	Grund	Folge	Action
E1802	Inputsource cannot be connected.	Fehler beim Empfangen des Dokumentes.	Empfang des momentan verarbeiteten Dokumentes vom Server wird abgebrochen und das nächste Dokument wird vom Server geholt. Das fehlerhafte Dokument wird beim nächsten Aufruf des Receive Tasks erneut vom Server geholt.	
E1803	Stream cannot be closed.	Fehler beim Schliessen des empfangenen Dokumentes.	Empfang des momentan verarbeiteten Dokumentes vom Server wird abgebrochen und das nächste Dokument wird vom Server geholt. Das fehlerhafte Dokument wird beim nächsten Aufruf des Receive Tasks erneut vom Server geholt.	
E2301	Try to update partner failed.	Sammelfehlermeldung. Grund siehe Untermeldung	Das PartnerUpdate wird abgebrochen und erst beim nächsten Start des UpdatePartnerJobs neu versucht.	Entsprechend Untermeldung.
E2401	Using IOInterface failed.	Sammelfehlermeldung. Grund siehe Untermeldung	Dieses Dokument konnte nicht korrekt vom Server gelesen werden und wird deshalb ausgelassen. Es wird nun versucht, das nächste Dokument vom Server zu lesen.	Entsprechend Untermeldung.
E2402	Receiving document failed.	Sammelfehlermeldung. Grund siehe Untermeldung	Dieses Dokument konnte nicht korrekt vom Server gelesen werden und wird deshalb ausgelassen. Es wird nun versucht, das nächste Dokument vom Server zu lesen.	Entsprechend Untermeldung.
E2403	RECEIVE DOC FAILED <TRANSPORTLOG>	Sammelfehlermeldung. Grund siehe Untermeldung	Dieses Dokument konnte nicht korrekt vom Server gelesen werden und wird deshalb ausgelassen. Der Empfang von Dokumenten vom Server wird abgebrochen und erst beim nächsten Start des Receive Task fortgesetzt.	
E2404	RECEIVE DOC FAILED	Keine Verbindung zum Server	Dieses Dokument konnte nicht korrekt vom Server gelesen werden und wird deshalb ausgelassen. Der Empfang von Dokumenten vom Server wird abgebrochen und erst beim nächsten Start des Receive Task fortgesetzt.	
E2501	Failed to delete send control file.	Das SendControlFile konnte nicht gelöscht werden. Es besteht ein Problem mit den Zugriffsrechten.	Es kann kein neues SendControlFile erzeugt werden.	Kontrolle der Zugriffsrechte und manuelles Löschen des SendControlFiles
E2502	SEND DOC FAILED <TRANSPORTLOG>	Sammelfehlermeldung. Grund siehe Untermeldung	Dieses Dokument konnte nicht korrekt zum Server übertragen werden. Und wird deshalb in das Error Verzeichnis verschoben.	Entsprechend Untermeldung.
E2503	Document cannot be send. Try to archive document failed.	Das Dokument konnte nicht gesendet werden. Der Versuch, das Dokument in das Error Verzeichnis zu verschieben schlug fehl.	Das Dokument befindet sich weiterhin im Send Verzeichnis, wird jedoch nicht mehr gesendet.	Manuelles Löschen des Dokumentes und erneutes Senden.
E2504	Failed to delete processed document [<filename>]	Das Dokument wurde erfolgreich zum Server gesandt. Der Versuch es anschliessend zu löschen schlug jedoch fehl.	Das erfolgreich gesendete Dokument befindet sich weiterhin im Send Verzeichnis.	Manuelles Löschen des Dokumentes.
E2505	Wrong SendControlFile informations.	Programm interner Fehler		MediData informieren.
E2601	Unimplemented Methode called	Programm interner Fehler		MediData informieren.

Error Nr.	Meldung	Grund	Folge	Action
E2901	Try to create partnerInfoFile failed.	Zugriffsproblem auf die Datenbank beim Erstellen des Partnerverzeichnis.	Das Partnerverzeichnis wurde nicht neu erstellt. Bei der nächsten Ausführung des PartnerUpdate Tasks wird erneut versucht, das Partnerverzeichnis zu erstellen.	
E2902	Try to create partnerInfoFile failed.	Zugriffsproblem auf das Filesystem beim Erstellen des Partnerverzeichnis.	Das Partnerverzeichnis wurde nicht neu erstellt. Bei der nächsten Ausführung des PartnerUpdate Tasks wird erneut versucht, das Partnerverzeichnis zu erstellen.	
E2903	Getting documentlist response failed	Fehler beim holen der neuen Dokumentliste vom Server.	Es wird beim nächsten Ausführung des Receive Tasks erneut versucht, die neuen Dokumente vom Server zu holen.	
E2904	Reading documentlist response failed[Operation timed out: connect]	Es kann keine Netzwerkverbindung zum Server hergestellt werden	Es werden in diesem Durchlauf des ReceiveJobs keine weiteren Dokumente mehr vom Server geholt. Bereits geholte Dokumente wurden korrekt verarbeitet. Beim nächsten Durchlauf des ReceiveJobs wird wieder neu versucht, die ev. Noch vorhandenen Dokumente vom Server zu holen.	Falls dieser Fehler mehrmals nacheinander auftritt, muss der Grund für die Netzwerkverbindungsprobleme analysiert werden (falsche Konfig des Proxy, Server?)
E2905	Parsing documentlist response failed.	Das Parsen der Dokumentliste vom Server schlug fehl.	Es wird beim nächsten Ausführung des Receive Tasks erneut versucht, die neuen Dokumente vom Server zu holen.	
E2906	Reading document failed.	Das Lesen des Dokumentes vom Server schlug fehl.	Der Empfang dieses Dokumentes wird übersprungen. Es wird beim nächsten Ausführung des Receive Tasks erneut versucht, dieses Dokument vom Server zu holen.	
E2907	Reading server response stream failed(Operation timed out: connect)	Es kann keine Netzwerkverbindung zum Server hergestellt werden	Die Verarbeitung der zu sendenden Dokumente wird abgebrochen und die aktuelle position im SendControlFile wird in der DB gespeichert. Beim nächsten Start das SendControlFile Tasks wird an der gleichen Stelle weitergefahren.	Falls dieser Fehler mehrmals nacheinander auftritt, muss der Grund für die Netzwerkverbindungsprobleme analysiert werden (falsche Konfig des Proxy, Server?)
E2908	Parsing server response failed.	Fehler beim Parsen der Serverantwort	Je nachdem, welche Serverantwort es war.	
E2909	Building https request failed. Could not parse Sender EAN[<fehler>] for File[<filename>]	Die im zu sendenden Dokument gefundene EAN war keine Zahl.	Dieses Dokument konnte nicht korrekt zum Server gesendet werden und wird deshalb ins Error Verzeichnis verschoben. Es wird nun versucht, das nächste Dokument vom Server zu senden.	
E2910	Build https req failed. Could not retrieve data from DB [<fehler>] for File[<filename>]	Fehler beim Zugriff auf die Datenbank.	Dieses Dokument konnte nicht korrekt zum Server gesendet werden und wird deshalb ins Error Verzeichnis verschoben. Es wird nun versucht, das nächste Dokument vom Server zu senden.	
E2911	Building https request failed. Could not parse document[<fehler>] for File[<filename>]	Fehler beim Parsen des zu versendenden Dokumentes.	Dieses Dokument konnte nicht korrekt zum Server gesendet werden und wird deshalb ins Error Verzeichnis verschoben. Es wird nun versucht, das nächste Dokument vom Server zu senden.	
E2912	Open https-connection failed.			
E2913	Getting addressbook failed.	Fehler beim Parsen des Adressbooks vom Server.	Das Partnerverzeichnis wurde nicht neu erstellt. Bei der nächsten Ausführung des PartnerUpdate Tasks wird erneut versucht, das Partnerverzeichnis zu erstellen.	

Error Nr.	Meldung	Grund	Folge	Action
E2914	Getting addressbook failed[<error>]	Fehler beim Parsen des Adressbooks vom Server.	Das Partnerverzeichnis wurde nicht neu erstellt. Bei der nächsten Ausführung des PartnerUpdate Tasks wird erneut versucht, das Partnerverzeichnis zu erstellen.	
E2915	Reading addressbook response failed[<error>]	IOException: Reading addressbook response failed(Operation timed out: connect)	Das Partnerverzeichnis wurde nicht neu erstellt. Bei der nächsten Ausführung des PartnerUpdate Tasks wird erneut versucht, das Partnerverzeichnis zu erstellen.	
E2916	Parsing addressbook response failed[<error>]	Fehler beim Parsen des Adressbooks vom Server.	Das Partnerverzeichnis wurde nicht neu erstellt. Bei der nächsten Ausführung des PartnerUpdate Tasks wird erneut versucht, das Partnerverzeichnis zu erstellen.	
E2918	Posting document failed[<error>]	Das Dokument kann nicht an den Server gesandt werden.	Das Dokument bleibt im Eingangsverzeichnis	
E2919	Read addressbook response failed[<error>]	Lesen des neuen Partnerverzeichnisses vom Server schlug fehl.	Das Partnerverzeichnis wurde nicht neu erstellt. Bei der nächsten Ausführung des PartnerUpdate Tasks wird erneut versucht, das Partnerverzeichnis zu erstellen.	
E2920	Parse addressbook response failed[<error>]	Parsen des neuen Partnerverzeichnisses vom Server schlug fehl.	Das Partnerverzeichnis wurde nicht neu erstellt. Bei der nächsten Ausführung des PartnerUpdate Tasks wird erneut versucht, das Partnerverzeichnis zu erstellen.	
E2921	Try to update partner on local database failed[<error>]	Schreiben neuen Partnerverzeichnisses in die lokale Datenbank schlug fehl.	Das Partnerverzeichnis wurde nicht neu erstellt. Bei der nächsten Ausführung des PartnerUpdate Tasks wird erneut versucht, das Partnerverzeichnis zu erstellen.	
E2922	Invalid URL[<url>] throws [<error>]	Ungültige URL	Document wird nicht empfangen/gesendet/partnerupdate nicht durchgeführt.	Medidata Informieren.
E2923	Parsing addressbook response failed[<error>]		Das Partnerverzeichnis wurde nicht neu erstellt. Bei der nächsten Ausführung des PartnerUpdate Tasks wird erneut versucht, das Partnerverzeichnis zu erstellen.	
E2924	Posting document failed[<error>]	Das Dokument kann nicht an den Server gesandt werden.	Das Dokument wurde ins Error Verzeichnis verschoben.	
E2925	Configured SenderEAN[nnn] doesn't match SenderEAN[nnn] in document[xyz]	Falsche Sender-EAN im Dokument.	Das Dokument wurde ins Error Verzeichnis verschoben.	
E2926	Getting getDocumentStates response failed[<error>]	Fehler beim Abfragen des Dokumentenstatus		Erneuter Versuch später, wenn einmalig
E2927	Getting getDocumentStates XmlParserException[<error>]	Fehler beim Parsen der Antwort des Dokumentenstatus		Erneuter Versuch später, wenn einmalig
E2928	Reading getDocumentStates response failed[<error>]	Fehler beim Abfragen des Dokumentenstatus		Erneuter Versuch später, wenn einmalig
E2929	unknown GetDocListResponse	Fehler beim Abfragen des Dokumentenstatus		Erneuter Versuch später, wenn einmalig
E2930	checkServer empty	Verbindungsaufnahme mit server fehlerhaft	Partnerverzeichnis wird nicht aktualisiert	Erneuter Versuch später, wenn einmalig

Error Nr.	Meldung	Grund	Folge	Action
E2931	checkServer failed.	Verbindungsaufnahme mit server fehlerhaft	Partnerverzeichnis wird nicht aktualisiert	Erneuter Versuch später, wenn einmalig
E2932	Reading checkServer response failed[<error>]	Verbindungsaufnahme mit server fehlerhaft	Partnerverzeichnis wird nicht aktualisiert	Erneuter Versuch später, wenn einmalig
E3501	File[<filename>] has malformedURLException[<error>]	Die URL eines Dokumentes ist ungültig.	Zu versendendes Dokument kann nicht eingelesen werden und wird ins Fehlerverzeichnis verschoben.	
E5701	Reading xml document failed[<error>]	Einlesen des XML Dokumentes schlug fehl.	Siehe übergeordnete Fehlermeldung.	
E5702	Parsing xml document failed [<error>]	Parsen des XML Dokumentes schlug fehl.	Siehe übergeordnete Fehlermeldung.	
E5703	Validation SAXNotSupportedEx ception [<error>]	Der XML Parser unterstützt diese Validierungsmethode nicht.		MediData informieren.
E5704	Validation SAXNotRecognizedE xception [<error>]	Programm interner Fehler		MediData informieren.
E5705	Can't create instance of <name>	Das angegebene Objekt kann nicht erzeugt werden		Ev. Zuwenig Speicher. Applikation neu starten und bei wiederholtem Auftreten MediData informieren.
E5801	XML VALIDATION problem[<problem>]	Problem der XML Validierung	Je nach eingestelltem xml.validation.abort.level wird das zu versendende Dokument verworfen oder nicht.	
E5802	XML VALIDATION switch problem[<problem>]	Programm interner Fehler		MediData informieren.

## 12.4 Meldungen vom Log-Level FATAL (F0000 – F9999)

Fatal Nr.	Meldung	Grund	Folge	Action	Bemerkung
F0204	Current certificate not found.	Zertifikat für aktuellen Monat nicht gefunden	Wird mit Meldung F2904 angehalten.	Keystore mit aktuellen Zertifikaten installieren	Kann nur beim start der Applikation auftreten.
F0205	keystorefile[<file name>] doesn't exist.	Fehler beim Initialisieren des Keystore, das Keystore File wurde nicht gefunden.	Wird mit Meldung F2904 angehalten.	Kontrollieren, ob keystore am richtigen platz mit richtigem Namen und im Konfigfile richtig eingetragen ist.	Kann nur beim start der Applikation auftreten.
F0206	Keystore initialization failed	Fehler beim Initialisieren des Keystore,	Wird mit Meldung F2904 angehalten.	Kontrollieren, ob keystore am richtigen Platz mit richtigem Namen.	Kann nur beim start der Applikation auftreten.
F0501	Property value is not a number.	Kann MPCCommunicator Konfigurationsfile nicht verarbeiten.	Wird mit Meldung F2201 angehalten	Kontrollieren des Konfigurationsfiles	Kann nur beim start der Applikation auftreten.

Fatal Nr.	Meldung	Grund	Folge	Action	Bemerkung
F0502	The KEYSTOREPASSWORDFILE[config/_ks.mp] does not exist.	Das Keystore Passwortfile wurde nicht gefunden.	Wird mit Meldung F2201 angehalten	Das Keystorepasswortfile muss an der angegebenen Position zur Verfügung stehen.	
F0503	Configurationfile not found.	Konfigurationsfile nicht gefunden	Applikation wird angehalten	Konfigurationsfile neu erstellen	
F0504	Decrypt of KEYSTOREPASSWORDFILE[<filename>] failed.	Das Keystorepasswortfile konnte nicht entschlüsselt werden weil es wahrscheinlich korrupt ist.	Applikation wird angehalten.	Das Keystorepasswortfile muss an der angegebenen Position neu zur Verfügung gestellt werden. Eventuell ist ein neues Zertifikat mit zugehörendem Passwortfile von der MediData zu beziehen.	
F0601	Cannot save new logging configfile.	Neues Logging Konfigurationsfile kann nicht erstellt werden.	Es existiert kein gültiges Logkonfigurationsfile und die Applikation wird deshalb angehalten.	Erstellen eines gültigen Logkonfigfiles.	
F0801	Get DBconnection failed.	Verbindung zur DB nicht möglich	Es kann nicht mit der DB kommuniziert werden	Applikation neu starten und Medidata melden.	todo: momentan wird hier nicht abgebrochen sondern prog läuft weiter
F0901	DB DriverClass [<drivename>] not found.	Der Treiber für die Datenbank kann nicht gefunden werden.	Applikation wird angehalten	Medidata Kontaktieren	Kann nur beim Start der Applikation auftreten.
F1801	Initialization of XML-parser failed.	XML Parser nicht initialisierbar.	Applikation wird angehalten	Medidata kontaktieren	Dieser Fehler kann nur während der Entwicklung auftreten, wenn ein XML Parser benutzt wird, der nicht alle Features unterstützt oder wenn das jar file den Parser nicht enthält.
F2201	failed to start scheduler	Konfigurationsfehler	Applikation wird angehalten	je nach untermeldung	Kann nur beim Start oder Stop der Applikation auftreten.
F2202	can't create/make needed directories	Kann benötigte Verzeichnisse(send/receive/error/partner) nicht finden/erzeugen	Applikation wird nicht gestartet	Kontrolle der Konfiguration und der Zugriffsrechte	
F2203	don't have full access to needed directories	Applikation hat nicht volle Zugriffsrechte (read/write/dell) auf benötigte Verzeichnisse(send/receive/error/partner)	Applikation wird nicht gestartet	Kontrolle der Konfiguration und der Zugriffsrechte	
F2501	Can't create InputHandler	Der InputHandler kann nicht erzeugt werden	Applikation wird angehalten		Dieser Fehler kann nicht auftreten, da diese Exception nur im Interface definiert ist, beim verwendeten FileInputHandler jedoch nie ausgelöst wird.

Fatal Nr.	Meldung	Grund	Folge	Action	Bemerkung
F2904	SSL initialization failed. [	SSL initialisierung ist fehlgeschlagen.	Die Applikation wird umgehend angehalten		Kann nur beim Start der Applikation auftreten.
F3301	Creating RemoteClientClass failed	HttpClient instanz kann nicht erzeugt werden	Die Applikation wird umgehend angehalten	Medidata Kontaktieren	Dieser Fehler kann nur während der Entwicklung auftreten, wenn im Konfig eine ungültige RemoteClient Klasse definiert wurde.
F3302	Creating RemoteClientClass failed	HttpClient instanz kann nicht erzeugt werden	Die Applikation wird umgehend angehalten	Medidata Kontaktieren	Dieser Fehler kann nur während der Entwicklung auftreten, wenn im Konfig eine ungültige RemoteClient Klasse definiert wurde.
F4201	F4201: Decrypting file[<filename>] failed.[<error>]	Ungültiges oder korruptes Keystore Passwortfile	Die Applikation wird umgehend angehalten	Das Keystorepasswortfile muss an der angegebenen position neu zur Verfügung gestellt werden.	Kann nur beim Start der Applikation auftreten.
F5701	Unknown XmlSaxParser-Handler		Die Applikation wird umgehend angehalten	Medidata Kontaktieren	Dieser Fehler kann nur während der Entwicklung auftreten, wenn ein nicht definierter XmlSaxParser-Handler angefordert wird.
F5702	Creating xml parser failed.		Die Applikation wird umgehend angehalten	Medidata Kontaktieren	Dieser Fehler kann nur während der Entwicklung auftreten, wenn ein XML Parser benutzt wird, der nicht alle Features unterstützt oder wenn das jar file den Parser nicht enthält.
F5801	XML VALIDATION problem[<problem>]	Problem der XML Validierung	Je nach eingestelltem xml.validation.abort.level wird das zu versendende Dokument verworfen oder nicht.	Kontrolle des zu versendenden XML Dokumentes bezüglich einhaltung des XML Schemas.	

## 13. Anhang C: Konfiguration und Logging

### 13.1 MPC Konfigurationsdateien

#### 13.1.1 Konfigurationsdatei

In der Konfigurationsdatei werden alle notwendigen Informationen definiert. D.h. die Konfiguration erfolgt mittels Editieren dieser Datei. (ASCII Textdatei)

Informationen sind:

- EAN – Absender
- DN - Absender
- IP – Empfänger
- Konfiguration der Tasks / Task – Manager (Interval oder Unix-Crontabs)
- Pfad der Verzeichnisse (relativ oder absolut)
  - Send + Sendarchive
  - Receive, ReceiveTest
  - Error
  - Partnerverzeichnis
- Name und Pfad des Keystores (um das Einspielen des neuen Keystores zu erleichtern wird der letzte Monat des Vorjahres ebenfalls mitgeliefert so dass im Dezember des Vorjahres die Zertifikate für das nächste Jahr bereits eingespielt werden können)
- Name der Sende-Steuerdatei. Alle SendControl Dateien müssen mit diesem Namen beginnen und mit .xml enden. Je nach Betriebssystem ist dieser Eintrag CaseSensitiv.
  - SendControl

#### Default MediPort Communicator Konfigurationsdatei

```
#####  
# MediPort Communicator  
#-----  
#  
# MPCCommunicator configurationfile  
#  
# MPC Version      01.20.00  
# Date             28.07.2016  
# Author           SUM, SCA, DIC, HIH, STY  
#  
# Copyright (c) 2065 MediData AG, Inc. All rights reserved.  
#####  
# History:  
# V01.20.00 no change per default, new optional property  
mpcommunicator.log4j.triggerdelay  
# V01.19.00 new property xml.generator.pretty_print
```

```
# V01.18.00, V01.17.00, V01.16.00: No changes in config.
# V01.15.00 configuration of hostnames instead of ip addresses
# V01.14.00 No changes per default, new optional config. properties may be added
#   to check/fix intermediate/via EAN(s) in send task (see documentation)
# V01.13.00 New config. properties for eKARUS conversion
# V01.12.00 New config. properties for XSLT transformation of received docs.
# V01.10.00, V01.11.00: No changes in config.
# V01.09.00 New config parameters:
#   -- 3 parameters for generic tasks
#   -- CLIENT.n.SENDARCHIVE_DIR and 4 other parameters for archiving task
#   -- 7 CLIENT.n.DIR.m parameters and 3 other parameters for writesendcontrol task
# V01.08.00 New config parameters:
#   -- all jcifs.* properties for SMB file server access
# V01.05.00 New config parameters:
#   -- CLIENT.x.y
# V01.04.00 New config parameters:
#   -- PARSE_RECEIVER, RECEIVER_EANLIST, UNKNOWN_RECEIVER_SUBDIR
#   -- DOCS_STATE_OUTPUTDIR, DOCS_STATE_INITIAL_DAYS
#   -- mpcommunicator.getdocstate.pollinterval,
#   mpcommunicator.getdocstate.cron,
#   mpcommunicator.task.getdocstate.enable
#=====

#=====
# MINIMAL CONFIGURATION
#=====

# MediPort Communicator Sender EAN
mpcommunicator.sender.ean=2099988874091

# MediPort Communicator Sender DN
mediport.dn=uid=EAN2099988874091,ou=Test,ou=MediPortTestfuerMPCommunicator,ou=MediData Intern,o=medidata.ch

# MediPort Host Name or Host Address
#   Produktiver Server: mpw1.medidata.ch
#   Test Server       : mpw0.medidata.ch
mpcommunicator.mediport.ip=mpw1.medidata.ch

#####
# Certificate
#####
# Keystore
keystore.name=config/EANxxxxxxxxxxxxx_mpg.keystore
keystore.passwordfile.name=config/_ks.mp

#=====
# OPTIONAL CONFIGURATION
#=====

#####
# PFADKONFIGURATION
#####
```

```
### NEUE MULTI CLIENT (SENDER) AB V01.05.00

CLIENT.1.EAN=2099988874091
CLIENT.1.SEND_DIR=data/send
CLIENT.1.SENDARCHIVE_DIR=data/send/archive
CLIENT.1.RECEIVE_DIR=data/receive
CLIENT.1.RECEIVETEST_DIR=data/receive/test
CLIENT.1.ERROR_DIR=data/error
#CLIENT.1.SEND_DIR=smb://dataserver/testuser/data/send
#CLIENT.1.SENDARCHIVE_DIR=smb://dataserver/testuser/data/send/archive
#CLIENT.1.RECEIVE_DIR=smb://dataserver/testuser/data/receive
#CLIENT.1.RECEIVETEST_DIR=smb://dataserver/testuser/data/receive/test
#CLIENT.1.ERROR_DIR=smb://dataserver/testuser/data/error
CLIENT.1.DOCSTAT_DIR=data/docstatus
CLIENT.1.PARTNER_FILE=data/partner/partnerinfo.txt

#example sendcontrol-configurations for client 1
#CLIENT.1.DIR.1=data/client1/invoices1
#CLIENT.1.DIR.1.DOCATTR=Tiers_Payant
#CLIENT.1.DIR.1.DOCPRINTED=true
#CLIENT.1.DIR.1.DISTTYPE=0
#CLIENT.1.DIR.1.PRINTLANGUAGE=D
#CLIENT.1.DIR.1.TRUSTCENTEREAN=1234567890123
#CLIENT.1.DIR.1.ISPAPERINVOICE=false
#CLIENT.1.DIR.1.SCHEMAID=26020
#CLIENT.1.DIR.2=data/client1/invoices2
#CLIENT.1.DIR.2.DOCATTR=Tiers_Garant_Manuell

#CLIENT.2.EAN=2099988870017
#CLIENT.2.SEND_DIR=data/send
#CLIENT.2.SENDARCHIVE_DIR=data/send/archive
#CLIENT.2.RECEIVE_DIR=data/receive
#CLIENT.2.RECEIVETEST_DIR=data/receive/test
#CLIENT.2.ERROR_DIR=data/error
#CLIENT.2.DOCSTAT_DIR=data/docstatus
#CLIENT.2.PARTNER_FILE=data/partner/partnerinfo.txt

#example xslt transformation instructions for received documents
#CLIENT.2.XSLT.1.DOCTYPE=generalInvoiceResponse_410.xsd
#CLIENT.2.XSLT.1.XSLT=data/xslt/generalInvoiceResponse_410.xslt
#CLIENT.2.XSLT.2.DOCTYPE=generalInvoiceResponse_430.xsd
#CLIENT.2.XSLT.2.XSLT=data/xslt/generalInvoiceResponse_430.xslt

#####
# DOCS_STATE
#####

# get all the docs which are done since (today - DOCS_STATE_INITIAL_DAYS) or
# since the last time the process was started
DOCS_STATE_INITIAL_DAYS=100
DOCS_STATE_NBR_OD_ENTRIES=1000

#####
# SENDCONTROL
```

```
#####  
# MediPort Communicator Sender Steuerdatei (SendControl010_xyz.xml)  
# REMARK: !!! Case sensitive!!! SendControl010_ != sendcontrol010_  
mpcommunicator.sender.controlfile=SendControl  
  
#####  
# TASKS (SIMPLE)  
#####  
# REMARK: Minimal values for the intervals are:  
# - PartnerUpdate : 60 [minutes]  
# - Send : 5 [seconds]  
# - Receive : 10 [minutes]  
# - Getdocstate : 24 [hours]  
# - Archivedelete:  
# - delete task : 1 [hour]  
# - store duration : 1 [day]  
# - Writesendcontrol : 1 [minute]  
#####  
  
# MediPort Communicator Partnerverzeichnis update interval[m]  
mpcommunicator.partner.updateinterval=360  
  
# MediPort Communicator send poll interval[s]  
mpcommunicator.send.pollinterval=300  
  
# MediPort Communicator receive poll interval[m]  
mpcommunicator.receive.pollinterval=60  
  
# MediPort Communicator getdocstate poll interval[h]  
mpcommunicator.getdocstate.pollinterval=24  
  
# MediPort Communicator delete sent and archived xmls poll interval[h]  
mpcommunicator.archivedelete.pollinterval=48  
# MediPort Communicator send archive store duration of sent files before automatic  
deletion[days]  
mpcommunicator.archivedelete.maxtime=365  
  
# MediPort Communicator write sendcontrol-xmls using configurations in CLIENT.x.DIR.y  
poll interval[m]  
mpcommunicator.writesendcontrol.pollinterval=5  
  
#####  
# MEDIPORT SERVER  
#####  
  
# MediPort Port  
mpcommunicator.mediport.port=443  
  
#####  
# Proxy Server  
#####  
# REMARK: Only "Basic" Proxy Authentication is supported  
#####
```

```
# Use Proxy Server (true/false)
proxy.use=false

# Proxy Server Host Name or Host Address (192.168.1.122/192.168.3.51)
proxy.ip=1.1.1.1

# Proxy Server Port(8443/8100)
proxy.port=8443

# Use Proxy Server User/Password authentication (true/false)
proxy.auth.use=true

# Proxy Server authentication username
proxy.auth.username=name

# Proxy Server authentication password
proxy.auth.password=password

#####
# RMI Server
#####
# REMARK: If rmi.enable is set to true then all the
#         tasks are set to false and you can only execute
#         a task by calling its rmi interface.
#####
rmi.serverip=localhost
rmi.serverport=4445
rmi.servicename=MPCCommunicator-Server
rmi.enable=false

#=====
# ADVANCED CONFIGURATION
#=====

#####
# TASKS (CRON)
#####
# Cron Expressions:
# Seconds Minutes Hours Day-of-Month Month Day-of-Week
#
# Seconds(0 to 59)
# Minutes(0 to 59)
# Hours(0 to 23)
# Day-of-Month(0-31)
# Month(0-11 or JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP,
#      OCT, NOV and DEC)
# Day-of-Week(1-7(1 = Sunday) or SUN, MON, TUE, WED, THU,
#      FRI and SAT)
#
# ranges and/or lists : "MON-WED,SAT"
#
# Wild-cards (the '*' character) every possible value of
#      this field
#
# The '/' character can be used to specify increments to
```

```
# values. ('0/15'~'0,15,30,45' 3/20'~'3,23,43')
#
# The '?' means no specific value for day-of-month and
# day-of-week fields.
#
# The 'L' means last for the day-of-month and day-of-week fields
# ("6L" or "FRIL"~"the last friday of the month")
#
# Example1("10 0/5 * * * ?")every 5 minutes, at 10 seconds
# after the minute(i.e. 10:00:10 am, 10:05:10 am, etc.).
# Example2("0 30 10-13 ? * WED,FRI") at 10:30, 11:30,
# 12:30, and 13:30, on every Wednesday and Friday.
# Example3("0 0/30 8-9 * 5,20 ?") every half hour between
# the hours of 8 am and 10 am on the 5th and 20th of
# every month.
#####

# MediPort Communicator Partnerverzeichnis update cron
#mpcommunicator.partner.update.cron=0 0 10-13 ? * WED,FRI

# MediPort Communicator send cron
#mpcommunicator.send.cron=0 0/5 10-13 ? * WED,FRI

# MediPort Communicator receive cron
#mpcommunicator.receive.cron=0 0/20 10-13 ? * WED,FRI

# MediPort Communicator getdocstate cron
#mpcommunicator.getdocstate.cron=0 0 10 ? * WED,FRI

# MediPort Communicator delete sent and archived xmls cron
#mpcommunicator.archivedelete.cron=0 0/5 10-13 ? * WED,FRI

# MediPort Communicator write sendcontrol cron
#mpcommunicator.writesendcontrol.cron=0 0/5 10-13 ? * WED,FRI

#####
# TASKS (ENABLE)
#####
# REMARK: If you run the MPCommunicator as a windows service
# or a unix demon then you can disable the filestop
# and the consolestop tasks.
# If you run the MPCommunicator as an application
# then you should at least enable one of the two
# stoptask or you can't stop the application.
#####

# Enable (true) or disable(false) MP Communicator Tasks
mpcommunicator.task.send.enable=true
mpcommunicator.task.receive.enable=true
mpcommunicator.task.partnerupdate.enable=true
mpcommunicator.task.filestop.enable=true
mpcommunicator.task.consolestop.enable=false
mpcommunicator.task.getdocstate.enable=false
mpcommunicator.task.archive.enable=false
mpcommunicator.task.writesendcontrol.enable=true
```

```
#####
# MediPort Server
#####

# MediPort Server Servlet definitions
mediport.servlet.getadrbookinfo=GetAdrBookInfo?Sender=
mediport.servlet.getpartner=GetPartner?Sender=
mediport.servlet.postdocument=PostDocument?
mediport.servlet.getdoclist=GetDocumentList?Sender=
mediport.servlet.getdocument=GetDocument?Sender=
mediport.servlet.setstatus=SetStatus?Sender=

#####
# n generic extension tasks
# extension jobs must be numbered consecutively starting from 1,
# and either the pollinterval or the cron property must be set
#####
# Plug-in extension tasks: if a class is specified, the task is also enabled
#extension.1.job=ch.md.mpc.mpc.ext.exampleclient.myjob
# Extension poll interval [minutes]
#extension.1.pollinterval=20
# Extension cron
#extension.1.cron=0 0/1 10-13 ? * WED,FRI

#extension.2.job=...
#extension.2.pollinterval=... (in minutes)
#extension.2.cron=...

#####
# SMB file server access for send, error and receive dirs.,
# i.e. Microsoft Windows "share" dirs.
#####
# For details, cf. http://jcifs.samba.org/src/docs/api/ for list of all 50+ jcifs
properties that may be set
# The default username used if not specified in an SMB URL:
#jcifs.smb.client.username=auser
# The default password used if not specified in an SMB URL:
#jcifs.smb.client.password=apassword
# The default authentication domain used if not specified in an SMB URL:
#jcifs.smb.client.domain
# The IP address of the WINS server. This is only required when accessing hosts on
different subnets although it is recommended if a WINS server is provided.
#jcifs.netbios.wins
# The local network's broadcast address. It may be necessary to set this for certain
network configurations because the default of 255.255.255.255 may otherwise throw a
"Network is unreachable" IOException. For example if the local host's IP address is
192.168.1.15, the broadcast address would likely be 192.168.1.255.
#jcifs.netbios.baddr
# This is extremely rare but NetBIOS provides for a "scope id" to be used in a
attempt to conceal groups of machines on the same network. Ask your network
administrator if scope id is used. If so, it must be set using this property or name
queries will fail.
#jcifs.netbios.scope
```

```
# The IP address of the local interface the client should bind to if it is different
from the default. For example if the client is to be used over a dial-up connection
the IP address of the PPP interface may need to be specified with this property.
#jcifs.smb.client.laddr
# The IP address of the local interface the client should bind to for name queries if
it is different from the default.
#jcifs.netbios.laddr
# The path to an lmhosts file containing a map of IP addresses to hostnames. The
format of this file is identical to that of the Windows lmhosts file format.
#jcifs.netbios.lmhosts

#-----
# XML Generator options
#-----
xml.generator.pretty_print=false
```

## 13.1.2 Konfiguration für die XML Schemaprüfung (Validierung)

```
=====
# MediPort Communicator configuration file for XML receive validation
#-----
# Replace GLN number with your GLN, add a separate set of properties
# for every GLN you use
#=====

#-----
# Turn on/off XML validation globally for this GLN
# - if false: overwrites the schema-specific switches
# - if true: the specific switches overwrite the global switch, if
#   available, otherwise true
#-----
mpcommunicator.receive.validation.enable=false

#-----
# Turn on/off XML validation individually for each schema
# - if the global switch is true, then these properties overwrite the global switch
#-----
#mpcommunicator.receive.validation.LabInvoiceRequest_300=true
#mpcommunicator.receive.validation.MDInvoiceRequest_300=true
#mpcommunicator.receive.validation.HospitalInvoiceRequest_300=true
#mpcommunicator.receive.validation.InvoiceReminderRequest_300=true
#mpcommunicator.receive.validation.MDInvoiceRequest_400=true
#mpcommunicator.receive.validation.HospitalInvoiceRequest_400=true
#mpcommunicator.receive.validation.PharmacyInvoiceRequest_400=true
#mpcommunicator.receive.validation.InvoiceReminderRequest_400=true
#mpcommunicator.receive.validation.HospitalCreditRequest_410=true
#mpcommunicator.receive.validation.GeneralCreditRequest_410=true
#mpcommunicator.receive.validation.GeneralInvoiceRequest_410=true
#mpcommunicator.receive.validation.XmitDocuments_400=true
#mpcommunicator.receive.validation.XmitDocuments_410=true
#mpcommunicator.receive.validation.GeneralInvoiceRequest_430=true
#mpcommunicator.receive.validation.HospitalMcdRequest_430=true
#mpcommunicator.receive.validation.StatusRequest_430=true
#mpcommunicator.receive.validation.GeneralContainer_430=true
#mpcommunicator.receive.validation.GeneralCreditRequest_430=true
#mpcommunicator.receive.validation.GeneralNotification_430=true
#mpcommunicator.receive.validation.GeneralFormRequest_430=true
```

```
#mpcommunicator.receive.validation.GeneralFormResponse_430=true
#mpcommunicator.receive.validation.Ekarus_x230=true
#mpcommunicator.receive.validation.Ekarus_x240=true
#mpcommunicator.receive.validation.GeneralInvoiceRequest_440=true
#mpcommunicator.receive.validation.HospitalMcdRequest_440=true
#mpcommunicator.receive.validation.GeneralNotification_440=true

#-----
# Turn on/off response generator globally
#-----
mpcommunicator.receive.validation.responsegenerator.enable=false

#-----
# Response generator
#-----
mpcommunicator.receive.validation.responsegenerator.name=MediPort Communicator V1.19
# version must be an integer >= 100:
mpcommunicator.receive.validation.responsegenerator.version=119

#-----
# Contact Data
#-----
mpcommunicator.receive.validation.responsegenerator.contact.companyname=company XYZ
mpcommunicator.receive.validation.responsegenerator.contact.department=department K&S
mpcommunicator.receive.validation.responsegenerator.contact.zip=6000
mpcommunicator.receive.validation.responsegenerator.contact.city=Musterstadt
mpcommunicator.receive.validation.responsegenerator.contact.email=abc@def.ch
mpcommunicator.receive.validation.responsegenerator.contact.phone=012 345 67 89
```

### 13.1.3 Konfiguration für die Adressierungskorrektur

```
#####
# MediPort Communicator configuration file for overwriting of XML
# receive recipient/insurance data
#-----
# Replace GLN number with your GLN, add a separate set of properties for
# every GLN you use
#####

#-----
# Turn on/off overwriting of recipient GLN
#-----
mpcommunicator.receive.modification.override_recipient_ean.enable=false

#-----
# Recipient GLN
#-----
mpcommunicator.receive.modification.recipient_ean=2000000000099

#-----
# Turn on/off overwriting of insurance data
#-----
mpcommunicator.receive.modification.override_insurance_data.enable=false

#-----
# Insurance data
#-----
```

```
# EAN - max. characters: 13
mpcommunicator.receive.modification.insurance_ean=2000000000077
# Company name - max. characters: 35
mpcommunicator.receive.modification.insurance_company_name=Muster Versicherung
# Company street - max. characters: 35
mpcommunicator.receive.modification.insurance_street=Musterstrasse 20
# Company zip - max. characters: 9
mpcommunicator.receive.modification.insurance_zip=6000
# Company city - max. characters: 35
mpcommunicator.receive.modification.insurance_city=Musterstadt

#-----
# Schemes for recipient
#-----
#mpcommunicator.receive.modification.recipient.MDInvoiceRequest_300=false
#mpcommunicator.receive.modification.recipient.InvoiceReminderRequest_300=false
#mpcommunicator.receive.modification.recipient.LabInvoiceRequest_300=false
#mpcommunicator.receive.modification.recipient.HospitalInvoiceRequest_300=false
#mpcommunicator.receive.modification.recipient.PharmacyInvoiceRequest_400=false
#mpcommunicator.receive.modification.recipient.MDInvoiceRequest_400=false
#mpcommunicator.receive.modification.recipient.InvoiceReminderRequest_400=false
#mpcommunicator.receive.modification.recipient.HospitalInvoiceRequest_400=false
#mpcommunicator.receive.modification.recipient.GeneralInvoiceRequest_410=false
#mpcommunicator.receive.modification.recipient.GeneralCreditRequest_410=false
#mpcommunicator.receive.modification.recipient.HospitalCreditRequest_410=false
#mpcommunicator.receive.modification.recipient.GeneralInvoiceRequest_430=false
#mpcommunicator.receive.modification.recipient.HospitalMcdRequest_430=false
#mpcommunicator.receive.modification.recipient.StatusRequest_430=false
#mpcommunicator.receive.modification.recipient.GeneralContainer_430=false
#mpcommunicator.receive.modification.recipient.GeneralCreditRequest_430=false
#mpcommunicator.receive.modification.recipient.GeneralNotification_430=false
#mpcommunicator.receive.modification.recipient.GeneralFormRequest_430=false
#mpcommunicator.receive.modification.recipient.GeneralFormResponse_430=false
#mpcommunicator.receive.modification.recipient.Ekarus_x230=false
#mpcommunicator.receive.modification.recipient.Ekarus_x240=false
#mpcommunicator.receive.modification.recipient.GeneralInvoiceRequest_440=false
#mpcommunicator.receive.modification.recipient.HospitalMcdRequest_440=false
#mpcommunicator.receive.modification.recipient.GeneralNotification_440=false

#-----
# Schemes for insurance
#-----
#mpcommunicator.receive.modification.insurance.MDInvoiceRequest_300=false
#mpcommunicator.receive.modification.insurance.InvoiceReminderRequest_300=false
#mpcommunicator.receive.modification.insurance.LabInvoiceRequest_300=false
#mpcommunicator.receive.modification.insurance.HospitalInvoiceRequest_300=false
#mpcommunicator.receive.modification.insurance.PharmacyInvoiceRequest_400=false
#mpcommunicator.receive.modification.insurance.MDInvoiceRequest_400=false
#mpcommunicator.receive.modification.insurance.InvoiceReminderRequest_400=false
#mpcommunicator.receive.modification.insurance.HospitalInvoiceRequest_400=false
#mpcommunicator.receive.modification.insurance.GeneralInvoiceRequest_410=false
#mpcommunicator.receive.modification.insurance.GeneralCreditRequest_410=false
#mpcommunicator.receive.modification.insurance.HospitalCreditRequest_410=false
#mpcommunicator.receive.modification.insurance.GeneralInvoiceRequest_430=false
#mpcommunicator.receive.modification.insurance.HospitalMcdRequest_430=false
#mpcommunicator.receive.modification.insurance.GeneralContainer_430=false
#mpcommunicator.receive.modification.insurance.GeneralCreditRequest_430=false
```

```
#mpcommunicator.receive.modification.insurance.GeneralNotification_430=false
#mpcommunicator.receive.modification.insurance.GeneralFormRequest_430=false
#mpcommunicator.receive.modification.insurance.Ekarus_x230=false
#mpcommunicator.receive.modification.insurance.Ekarus_x240=false
#mpcommunicator.receive.modification.insurance.GeneralInvoiceRequest_440=false
#mpcommunicator.receive.modification.insurance.HospitalMcdRequest_440=false
#mpcommunicator.receive.modification.insurance.GeneralNotification_440=false
```

## 13.1.4 Windows Service Konfigurationsfile

Wenn die Applikation als Service konfiguriert wird, wird das Java Service Wrapper Konfigurationsfile benötigt.

### Java Service Wrapper Konfigurationsfile

```
*****
# Wrapper parameters for MP Communicator
*****
# Java Application
wrapper.java.command=./jre/bin/java

# Java Main class
wrapper.java.mainclass=com.medidata.mediport.communicator.service.CommunicatorService
Wrapper

# Java Classpath (include wrapper.jar) Add class path elements as
# needed starting from 1
wrapper.java.classpath.1=./lib/MPCCommunicator.jar
wrapper.java.classpath.2=./lib/iaik_jce_full_signed.jar
wrapper.java.classpath.3=./lib/MPCCommunicator_libs.jar

# Java Library Path (location of Wrapper.DLL or libwrapper.so)
wrapper.java.library.path=./lib
# Java Additional Parameters
#wrapper.java.additional.1=
# Initial Java Heap Size (in MB)
wrapper.java.initmemory=3
# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=128
# Application parameters. Add parameters as needed starting from 1
wrapper.app.parameter.1=./
# Port which the native wrapper code will attempt to connect to
wrapper.port=1777

*****
# Wrapper Logging parameters
*****
# Format of output for the console. (See docs for formats)
wrapper.console.format=PM

# Log Level for console output. (NONE, STATUS, DEBUG, INFO, ERROR, FATAL)
wrapper.console.loglevel=NONE
# Log file to use for wrapper output logging.
wrapper.logfile=./data/log/wrapper.log
# Format of output for the log file. (See docs for formats)
wrapper.logfile.format=LPTM
```

```
# Log Level for log file output. (NONE, STATUS, DEBUG, INFO, ERROR, FATAL)
wrapper logfile.loglevel=STATUS
# Maximum size that the log file will be allowed to grow to before
# the log is rolled. Size is specified in bytes. The default value
# of 0, disables log rolling. May abbreviate with the 'k' (kb) or
# 'm' (mb) suffix. For example: 10m = 10 megabytes.
wrapper logfile.maxsize=1m

# Maximum number of rolled log files which will be allowed before old
# files are deleted. The default value of 0 implies no limit.
wrapper logfile.maxfiles=1

# Log Level for sys/event log output. (See docs for log levels)
wrapper syslog.loglevel=NONE

*****
# Advanced properties
*****
# Number of seconds to allow between the time that the JVM reports
# that it is stopped and the time that the JVM process actually
# terminates. 0 means never time out. Defaults to 5 seconds.
wrapper jvm_exit.timeout=180

# Number of seconds to allow between the time that the Wrapper
# asks the JVM to shutdown and the time that the JVM side of the
# Wrapper responds that it is stopping. 0 means never time out.
# Defaults to 30 seconds.
wrapper shutdown.timeout=180

*****
# Wrapper NT Service parameters READ THE !!!WARNING!!!
*****
# !!!WARNING!!! - Do not modify any of these parameters when an application
# using this configuration file has been installed as a service.
# Please uninstall the service before modifying this section. The
# service can then be reinstalled.

# Name of the service
wrapper ntservice.name=MediData MediPort Communicator

# Display name of the service
wrapper ntservice.displayname=MediData MediPort Communicator
# Description of the service
wrapper ntservice.description=MediData MediPort Communicator
# Service dependencies. Add dependencies as needed starting from 1
wrapper ntservice.dependency.1=

# Mode in which the service is installed. AUTO_START or DEMAND_START
wrapper ntservice.starttype=AUTO_START

# Priority at which the service is run. NORMAL, LOW, HIGH, or
# REALTIME
wrapper ntservice.process_priority=NORMAL

# Allow the service to interact with the desktop.
wrapper ntservice.interactive=false
# !!!WARNING!!! - Do not modify any of these parameters when an application
```

```
# using this configuration file has been installed as a service.  
# Please uninstall the service before modifying this section. The  
# service can then be reinstalled.
```

## 13.2 MPCommunicator Logging

### 13.2.1 Log Konfigurationsfile

#### Default MediPort Communicator LogKonfigurationsdatei

```
##### DEFINE LOGGERS  
log4j.rootCategory=INFO, A1, A2  
  
##### FILE APPENDER A1  
log4j.appender.A1=org.apache.log4j.RollingFileAppender  
log4j.appender.A1.File=data/log/mpcommunicator.log  
log4j.appender.A1.MaxFileSize=5000KB  
log4j.appender.A1.MaxBackupIndex=3  
log4j.appender.A1.layout=org.apache.log4j.PatternLayout  
log4j.appender.A1.layout.ConversionPattern=%d{ISO8601} %-1x %-5p - %m%n  
  
##### CONSOLE APPENDER A2  
log4j.appender.A2=org.apache.log4j.ConsoleAppender  
log4j.appender.A2.layout=org.apache.log4j.PatternLayout  
log4j.appender.A2.layout.ConversionPattern=%d{ISO8601} %-1x %-5p - %m%n  
  
##### Email: SMTP APPENDER EMAIL  
log4j.appender.EMAIL=org.apache.log4j.net.SMTPAppender  
log4j.appender.EMAIL.Threshold=ERROR  
log4j.appender.EMAIL.SMTPHost=mail.testdomain.ch  
#the following 5 properties are used e.g. for gmail: smtps is for SSL:  
#log4j.appender.EMAIL.SMTPProtocol=smtps  
#log4j.appender.EMAIL.SMTPPort=465  
#log4j.appender.EMAIL.smtp.starttls.enable=true  
#log4j.appender.EMAIL.SMTPUsername=myuser@mydomain.com  
#log4j.appender.EMAIL.SMTPPassword=mypw  
log4j.appender.EMAIL.From=mpc-test@testdomain.ch  
log4j.appender.EMAIL.To=mpc-test@testdomain.ch  
log4j.appender.EMAIL.Subject=MPCommunicator Log Message  
log4j.appender.EMAIL.BufferSize=1  
#to change the default triggering criteria (send email for every ERROR+ event), can  
#specify a class that implements org.apache.log4j.spi.TriggeringEventEvaluator:  
log4j.appender.EMAIL.EvaluatorClass=com.medidata.mediport.communicator.utility.Log4jT  
riggerAfterDelay  
log4j.appender.EMAIL.layout=org.apache.log4j.PatternLayout  
log4j.appender.EMAIL.layout.ConversionPattern=%d{ISO8601} %-1x %-5p - %m%n  
#log4j.appender.EMAIL.SMTPDebug=true  
  
##### SNMP APPENDER  
log4j.appender.SNMP=org.apache.log4j.ext.SNMPTrapAppender  
# ImplementationClassName: Set the value of the concrete class that implements  
# the SnmpTrapSenderFacade interface.  
log4j.appender.SNMP.ImplementationClassName=org.apache.log4j.ext.JoeSNMPTrapSender  
log4j.appender.SNMP.layout=org.apache.log4j.PatternLayout  
log4j.appender.SNMP.layout.ConversionPattern=%d %-5p - %m%n
```

```
# ManagementHost: Set the IP address of the remote host that traps should be
# sent to.
log4j.appender.SNMP.ManagementHost=pcnsum
# ManagementHostTrapListenPort: Set the port used on the remote host to listen
# for SNMP traps. The standard is 162.
log4j.appender.SNMP.ManagementHostTrapListenPort=162
# EnterpriseOID: Set the enterprise OID that will be sent in the SNMP PDU.
log4j.appender.SNMP.EnterpriseOID=1.3.6.1.4.1.24.0
# LocalIPAddress: Set the IP address of the host that is using this appender
# to send SNMP traps. This address will be encoded in the SNMP PDU, and used to
# provide things like the "agent"'s IP address.
log4j.appender.SNMP.LocalIPAddress=192.168.1.99
# LocalTrapSendPort: Set the value of the port that will be used to send traps
# out from the local host.
log4j.appender.SNMP.LocalTrapSendPort=163
# GenericTrapType: Set the generic trap type for this SNMP PDU. The allowed
# values for this attribute are a part of the SNMP standard:
# 0 -- cold start
# 1 -- warm start
# 2 -- link down
# 3 -- link up
# 4 -- authentication failure
# 5 -- EGP neighbor loss
# 6 -- enterprise specific
log4j.appender.SNMP.GenericTrapType=6
# SpecificTrapType: Set the specific trap type for this SNMP PDU. Can be used
# for application and/or enterprise specific values. any value within the range
# defined for an INTEGER in the ASN.1/BER notation; i.e. -128 to 127
log4j.appender.SNMP.SpecificTrapType=0
# ApplicationTrapOID: Set the trap OID that will be sent in the SNMP PDU for
# this app.
log4j.appender.SNMP.ApplicationTrapOID=1.3.6.1.4.1.24.2
# CommunityString: Set the community string set for the SNMP session this
# appender will use. The community string is used by SNMP (prior to v.3) as a
# sort of plain-text password.
log4j.appender.SNMP.CommunityString=public
# Set the value of the system up time that will be used for the SNMP PDU.
log4j.appender.SNMP.SysUpTime=1000
```

## 13.2.2 MPC Logfile

Nachfolgend ein Beispiels - MP Communicator Logfile. Es ist dabei zu beachten, dass im Originallogfile kein „Word Wrapping“ stattfindet und deshalb jede Zeile immer mit dem Datumseintrag beginnt.

### MPCommunicator Logfile

```

2003-03-21 16:09:25,187 M INFO - I2201: *****
2003-03-21 16:09:25,187 M INFO - I2201: *** Start MediPort Communicator ***
2003-03-21 16:09:25,187 M INFO - I2201: ***
2003-03-21 16:09:25,187 M INFO - I2201: *** to stop:
2003-03-21 16:09:25,187 M INFO - I2201: *** exit<ret> or Put mpcommunicator_stop file ***
2003-03-21 16:09:25,234 M INFO - I2201: *** into data directory. ***
2003-03-21 16:09:25,234 M INFO - I2201: *****
2003-03-21 16:09:25,234 M INFO - I2201: Starting MPCommunicator V1.0.0
2003-03-21 16:09:25,453 M INFO - I2204: ----- Scheduler schedule jobs -----
2003-03-21 16:09:25,515 M INFO - I2205: grp_mg.job_PartnerUpdate will run at[Fri Mar 21 16:09:30 CET 2003] & repeat[-1/7200000]
2003-03-21 16:09:25,531 M INFO - I2205: grp_mg.job_Receive will run at[Fri Mar 21 16:09:30 CET 2003] & repeat[-1/60000]
2003-03-21 16:09:25,531 M INFO - I2205: grp_mg.job_SendControlFile will run at[Fri Mar 21 16:09:30 CET 2003] & repeat[-1/10000]
2003-03-21 16:09:25,531 M INFO - I2205: grp_mg.job_ConsoleStop will run at[Fri Mar 21 16:09:30 CET 2003] & repeat[-1/10000000000]
2003-03-21 16:09:25,531 M INFO - I2206: ----- Scheduler started -----
2003-03-21 16:09:30,625 P INFO - I2904: Current cert = 03-2003
2003-03-21 16:09:31,718 S INFO - I1001: Start processing new SendControlFile[NbrOfDocs[8] DocsProcessed[0] SendfileSize[1883]
LastModified[Thu Feb 27 16:16:07 CET 2003]]
2003-03-21 16:09:37,593 P INFO - I2902: ADRBOOK UNCHANGED.
2003-03-21 16:09:37,625 P INFO - I0801: New partnerfile[.data\partner\partnerinfo.txt] created.
2003-03-21 16:09:55,734 S INFO - I2501: SEND DOC OK FILENAME[.data\send\HOi_0001.xml]
RECEIVERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[100] MDID[5336]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259375312]
2003-03-21 16:09:56,640 R INFO - I2401: RECEIVE DOC OK FILENAME[.data\receive\test\IRHSP03001048259396187TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5323]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:13,921 S INFO - I2501: SEND DOC OK FILENAME[.data\send\HOi_0001a.xml]
RECEIVERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[100] MDID[5337]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259375312]
2003-03-21 16:10:14,953 R INFO - I2401: RECEIVE DOC OK FILENAME[.data\receive\test\IRHSP03001048259414484TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5324]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:16,859 S INFO - I2501: SEND DOC OK FILENAME[.data\send\HOi_0002.xml]
RECEIVERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[100] MDID[5338]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259375312]
2003-03-21 16:10:17,703 R INFO - I2401: RECEIVE DOC OK FILENAME[.data\receive\test\IRHSP03001048259417218TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5325]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:20,390 S INFO - I2501: SEND DOC OK FILENAME[.data\send\HOi_0002a.xml]
RECEIVERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[100] MDID[5339]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259375312]
2003-03-21 16:10:21,437 R INFO - I2401: RECEIVE DOC OK FILENAME[.data\receive\test\IRHSP03001048259420968TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5326]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:23,203 S INFO - I2501: SEND DOC OK FILENAME[.data\send\HOi_0003.xml]
RECEIVERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[100] MDID[5340]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259375312]
2003-03-21 16:10:24,125 R INFO - I2401: RECEIVE DOC OK FILENAME[.data\receive\test\IRHSP03001048259423656TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5327]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:26,265 S INFO - I2501: SEND DOC OK FILENAME[.data\send\HOi_0003a.xml]
RECEIVERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[100] MDID[5341]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259375312]

```

```

2003-03-21 16:10:27,046 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259426625TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5328]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:28,875 S INFO - I2501: SEND DOC OK FILENAME[.\\data\\send\\H0rem_0001.xml]
RECEIVERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[100] MDID[5342]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSD InvoiceReminderRequest_300.xsd] JOB[1048259375312]
2003-03-21 16:10:29,734 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259429234TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5329]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:30,875 S INFO - I2501: SEND DOC OK FILENAME[.\\data\\send\\H0rem_0006.xml]
RECEIVERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[100] MDID[5343]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSD InvoiceReminderRequest_300.xsd] JOB[1048259375312]
2003-03-21 16:10:31,734 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259431250TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5330]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:32,562 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259432093TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5331]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:33,468 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRREX03001048259433031TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5332]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDInvoiceReminderRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:34,375 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRREX03001048259433906TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5333]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDInvoiceReminderRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:35,218 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259434734TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5334]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:10:36,046 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259435578TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5335]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259370109]
2003-03-21 16:11:31,578 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259491125TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5336]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259490109]
2003-03-21 16:11:32,625 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259492125TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5337]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259490109]
2003-03-21 16:11:33,421 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259492953TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5338]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259490109]
2003-03-21 16:11:34,281 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259493781TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5339]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259490109]
2003-03-21 16:11:35,109 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259494625TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5340]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259490109]
2003-03-21 16:11:35,906 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRHSP03001048259495437TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5341]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDHospitalInvoiceRequest_300.xsd] JOB[1048259490109]
2003-03-21 16:11:36,828 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRREX03001048259496375TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5342]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDInvoiceReminderRequest_300.xsd] JOB[1048259490109]
2003-03-21 16:11:37,687 R INFO - I2401: RECEIVE DOC OK FILENAME[.\\data\\receive\\test\\IRREX03001048259497250TMP0.XML]
SENDERDN[uid=EAN2099988870017,ou=Test,ou=MPGateway,ou=Kostentraeger,o=medidata.ch] STATETYPEID[200] MDID[5343]
DOCTYPE[http://www.xmlData.ch/xmlInvoice/XSDInvoiceReminderRequest_300.xsd] JOB[1048259490109]
2003-03-21 16:12:30,578 R INFO - I2903: NO NEW DOCUMENTS ON SERVER.
2003-03-21 16:13:20,171 M INFO - I2207: ----- Shutting Down -----
2003-03-21 16:13:20,171 M INFO - I2203: RunningSince[Fri Mar 21 16:09:25 CET 2003] NbrOfJobsExecuted[29] NbrDocs Sent[8]
NotSent[0] Received[21]
2003-03-21 16:13:20,171 M INFO - I2208: ----- Shutdown Complete -----

```

## 14. Anhang D: Scheduler Konfiguration

### 14.1 Cron Expressions

Cron-Expressions are used to configure instances of CronTrigger. Cron-Expressions are strings that are actually made up of six sub-expressions, that describe individual details of the schedule. These sub-expression are separated with white-space, and represent:

Seconds

Minutes

Hours

Day-of-Month

Month

Day-of-Week

An example of a complete cron-expression is the string "0 0 12 ? \* WED" - which means "every Wednesday at 12:00 pm".

Individual sub-expressions can contain ranges and/or lists. For example, the day of week field in the previous (which reads "WED") example could be replaced with "MON-FRI", "MON, WED, FRI", or even "MON-WED,SAT".

Wild-cards (the '\*' character) can be used to say "every" possible value of this field. Therefore the '\*' character in the "Month" field of the previous example simply means "every month". A '\*' in the Day-Of-Week field would obviously mean "every day of the week".

All of the fields have a set of valid values that can be specified. These values should be fairly obvious - such as the numbers 0 to 59 for seconds and minutes, and the values 0 to 23 for hours. Day-of-Month can be any value 0-31, but you need to be careful about how many days are in a given month! Months can be specified as values between 0 and 11, or by using the strings JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV and DEC. Days-of-Week can be specified as values between 1 and 7 (1 = Sunday) or by using the strings SUN, MON, TUE, WED, THU, FRI and SAT.

The '/' character can be used to specify increments to values. For example, if you put '0/15' in the Minutes field, it means 'every 15 minutes, starting at minute zero'. If you used '3/20' in the Minutes field, it would mean 'every 20 minutes during the hour, starting at minute three' - or in other words it is the same as specifying '3,23,43' in the Minutes field.

The '?' character is allowed for the day-of-month and day-of-week fields. It is used to specify "no specific value". This is useful when you need to specify something in one of the two fields, but not the other. See the examples below (and CronTrigger JavaDOC) for clarification.

The 'L' character is allowed for the day-of-month and day-of-week fields. This character is short-hand for "last", but it has different meaning in each of the two fields. For example, the value "L" in the day-of-month field means "the last day of the month" - day 31 for January, day 28 for February on non-leap years. If used in the day-of-week field by itself, it simply means "7" or "SAT". But if used in the day-of-week field after another value, it means "the last xxx day of the month" - for example "6L" or "FRIL" both mean "the last friday of the month". When using the 'L' option, it is important not to specify lists, or ranges of values, as you'll get confusing results.

Here are a few more examples of expressions and their meanings - you can find even more in the JavaDOC for CronTrigger

CronTrigger Example 1 - an expression to create a trigger that simply fires every 5 minutes

```
"0 0/5 * * * ?"
```

CronTrigger Example 2 - an expression to create a trigger that fires every 5 minutes, at 10 seconds after the minute (i.e. 10:00:10 am, 10:05:10 am, etc.).

```
"10 0/5 * * * ?"
```

CronTrigger Example 3 - an expression to create a trigger that fires at 10:30, 11:30, 12:30, and 13:30, on every Wednesday and Friday.

```
"0 30 10-13 ? * WED,FRI"
```

CronTrigger Example 4 - an expression to create a trigger that fires every half hour between the hours of 8 am and 10 am on the 5th and 20th of every month. Note that the trigger will NOT fire at 10:00 am, just at 8:00, 8:30, 9:00 and 9:30

```
"0 0/30 8-9 * 5,20 ?"
```

## 15. Anhang E: Release Notes

Siehe Datei *ReleaseNotes.txt* unter dem Installationspfad.