



**MEDIATRIX  
RESPONSE**

Dokumentation

# Mediatrix Response / Mailroom

ENTERPRISE INPUT & RESPONSE MANAGEMENT



Powered by Intelligence

Autor	ITyX GmbH   Carl-Benz-Str. 10-12   DE-56218 Mülheim-Kärlich
Version	Version 0.91
Letzte Änderung	09.10.2015 17:24:17

  
**ITyX**  
GOOD TO KNOW.

## Inhaltsverzeichnis

1	Allgemeines .....	5
2	Systemanforderungen .....	7
2.1	Datenbank .....	7
2.2	Hardware .....	8
2.3	Client.....	10
2.4	Speicher .....	12
3	Grundinstallation des Systems .....	14
3.1	Datenbank-Setup.....	14
3.2	Installationsverzeichnis anlegen .....	14
3.3	Konfigurationsdateien anpassen.....	16
3.4	Datenbanken initialisieren .....	18
3.5	Registrierung der Software .....	18
3.6	Dienste installieren.....	23
3.6.1	Windows Dienste installieren.....	23
3.6.2	Linux Dienste installieren .....	24
3.7	Starten der Software .....	27
3.8	Clients starten .....	28
3.8.1	Contex Admin starten.....	28
3.8.2	Mediatrix-Client starten .....	28
4	Allgemeine Systemkonfiguration .....	30
4.1	Übersicht der Verzeichnisstruktur.....	30
4.2	Start-Scripts .....	32
4.2.1	/bin/.....	32
4.2.2	/bin/install/.....	33
4.2.3	/init-scripts/ .....	35
4.2.4	Update-Scripts .....	36
4.3	Kommunikation zwischen den Komponenten .....	36
4.3.1	Externe Kommunikation zwischen den Komponenten .....	36
4.3.2	Interne Kommunikation zwischen den Komponenten.....	36
4.3.3	Liste der Protokolle und Ports .....	38
4.3.4	Zuordnung JVM und Komponenten .....	44

4.4	Konfiguration mediatrix.properties.....	46
4.4.1	Datenbank-Einstellungen .....	46
4.4.2	Allgemeines .....	50
4.4.3	Single-Sign-On .....	79
4.4.4	Alternative FunctionChecker-Erweiterung.....	81
4.4.5	c3p0-Connection-Pool-Einstellungen .....	83
4.5	Mediatrix Authentifizierung .....	87
4.5.1	Unterschied zwischen den Authentifizierungsmethoden .....	88
4.5.2	Quick Guide .....	89
5	Systembetrieb .....	91
5.1	Tomcat Tuning.....	91
5.2	E-Mail-Verarbeitung (Monitoring).....	93
5.3	Logging .....	96
5.3.1	Konfiguration in Mediatrix .....	96
5.3.2	Konfiguration in Context.....	96
5.3.3	Logdateien der einzelnen Wrapper.....	97
5.4	Dokumenten-Zustände (Status) .....	97
5.5	Performance Monitoring.....	103

Mediatrix ist eine Response-Management-Software, die Nachrichten über unterschiedliche Kanäle (E-Mail, Fax, Brief, Facebook, Telefon etc.) verarbeiten kann. Die Software ist in Java programmiert und arbeitet auf Basis eines Tomcat-Applikationsservers. Die Software kann auf jedem Betriebssystem betrieben werden, auf der eine Java-Umgebung installiert ist. Sämtliche Daten sind zentral in einer Datenbank gespeichert, auf den Clients bleiben keinerlei Rückstände von Kundendaten oder -dokumente. Die Software kann on-demand über einen Web-Browser heruntergeladen werden, die einzige Prämisse ist eine Installation von Java auf der Workstation.

---

*Die Java-Version sollte auf Server und Clients identisch sein. D.h. es sollten keine Kombinationen von Java 1.7 und Java 1.8 entstehen, da es hier Inkompatibilitäten geben wird. Hinzu kommt, dass Mediatrix v2.0 und v2.1 nicht mit Java 1.7 kompatibel ist, die Software muss mit 1.6 gestartet werden. Mediatrix v2.5 setzt hingegen Java 1.7 oder 1.8 voraus.*

---

## 1 Allgemeines

Mediatrix-Response ist ein modulares Software-System, welches aus mehreren Diensten besteht. Kernstück der Software ist „Mediatrix WebApp Server“ (= Web-Application-Server). Mediatrix arbeitet auf Basis von „Apache Tomcat 8.0“, allerdings wurde die Verzeichnisstruktur derart verändert, dass neue Installationen von Mediatrix nicht in vorhandene Applikationsserverumgebungen integriert werden sollten. ITyX liefert ein fertiges Software-Paket aus, die einen Tomcat-Server<sup>1</sup> bereits enthält.

Mediatrix nutzt für die Beschreibung der Organisationsstruktur eigene Begriffe, die nun kurz angerissen werden - eine ausführliche Liste finden Sie im Glossar (Kapitel 5.5). Die Namen sind frei wählbar, die Hierarchie ist jedoch vorgegeben. Zusammenfassend hat Mediatrix mindestens ein „Projekt“, welches alle Nachrichten für ein Service-Team beinhalten sollte. Um Nachrichten zu sortieren bzw. zu klassifizieren werden diese in mindestens eins, meistens mehrere „Teilprojekte“ aufgeteilt. Teilprojekte sind stets genau einem Projekt zugewiesen, quasi ein Unterordner eines Projekts. Teilprojekte können zudem „Ordner“ haben, um Nachrichten weiter zu granulieren. Ordner sind jedoch nicht so präzise einstellbar wie ein Teilprojekt. Doch dazu später mehr in Kapitel **Fehler! erweisquelle konnte nicht gefunden werden.**

Neben dem Applikationsserver für Mediatrix werden weitere Stand-Alone-Programme mitgeliefert, welche „Zuarbeit“ für Mediatrix leisten. Jeder der hier aufgeführten Dienste kann eigenständig arbeiten, es besteht keine Abhängigkeit oder Notwendigkeit für eine Startreihenfolge<sup>2</sup>:

Dienst-Name	Beschreibung
WebApp Server	Kernstück der Mediatrix-Software-Suite. Dieser Dienst stellt den Mediatrix-Server zur Verfügung, über den Mediatrix-Clients heruntergeladen, gestartet und auf den neusten Stand gebracht werden können.
Emailedemon	Dieses Programm lädt E-Mails herunter, E-Mail-Konten werden innerhalb der Mediatrix-Projekte konfiguriert. Die E-Mails werden über diesen Dienst in der Datenbank gespeichert und so Mediatrix zur Verfügung gestellt. Der Mediatrix-Emailedemon beherrscht POP3 / POP3S und IMAP / IMAPS. Das Versenden von Nachrichten erfolgt nicht über Emailedemon, sondern direkt über den Mediatrix-WebApp-Server.
Outbound	Outbound versendet E-Mails, die über E-Mail-Listen generiert wurden. Sollen also Newsletter über Mediatrix verwaltet werden, ist dies der Dienst, über den sie versendet werden.

<sup>1</sup> Andere Applikationsserver wie Glassfish oder Websphere werden nicht unterstützt

<sup>2</sup> Der PTME-Dienst kann je nach Umfang der KI-Daten erst verzögert verfügbar sein, weswegen es unter Umständen empfehlenswert ist diesen Dienst vor allen anderen Diensten hochzufahren

Dienst-Name	Beschreibung
PTME	Der PTME-Dienst ist ein Vorgänger von Contex (einem weiteren ITyX-Produkt). Mithilfe des PTME-Dienstes werden eingehende E-Mails analysiert. Eine KI-Komponente erlernt das Verhalten der Mitarbeiter; Mails mit ähnlichem Inhalt werden durch die PTME automatisch in vordefinierte Teilprojekte geleitet. Dieses Vorgehen wird auch „Routing“ genannt.
Escalation	Eingehende Nachrichten können mithilfe des Escalation-Demons nach Eingang an eine SLA-Zeit (Service-Level-Agreement) geknüpft werden. Sollte eine Nachricht bis zum Ablauf dieser Zeit nicht bearbeitet worden sein, so kann innerhalb von Mediatrix eine Eskalation der Mail eingerichtet werden, die automatisch über diesen Dienst erfolgt. Eskalationen können bis zu drei Stufen haben, eskalierte Nachrichten werden farblich im Mediatrix-Client markiert und, wenn gewünscht, werden durch diesen Dienst Nachrichten an weitere Personen geschrieben, sobald eine Eskalations-Stufe erreicht wurde.
Report-Generator	Mediatrix kann aus der eigenen Datenbank Reports über eine Vielzahl an Themen generieren. Dies kann manuell über den Mediatrix-Client erfolgen oder über den Report-Generator-Demon. So ist es möglich einzelne Reports in festen Zeitintervallen zu definieren und diese entweder als Datei abzuspeichern (PDF, XLS oder CSV) bzw. per E-Mail zu verschicken (HTML).
POP3-EML-Server	Hierbei handelt es sich um einen sehr simplen POP3-Server, der aus einem bestimmten Verzeichnis *.eml-Dateien laden und in Mediatrix importieren kann. Verzeichnisse können über „Konten“ in der Mediatrix-Projektadministration eingerichtet werden.

Alle Dienste können mithilfe eines Java-Wrapper-Tools (Tanuki Wrapper) auch als Windows-Dienst bzw. Linux-Daemon installiert werden. Für Test-Zwecke können die Programme jedoch auch über Script-Dateien im `mediatrix-Contex/bin/`-Verzeichnis als Applikation gestartet werden.

## 2 Systemanforderungen

Mediatrix-Client nutzt als Speicherablage den Arbeitsspeicher der Workstations und die zugrunde liegende Datenbank. Auf den Workstations selbst werden nur ein Profil sowie einige temporäre Dateien gespeichert. Das führt dazu, dass alle Daten von und zur Datenbank gesendet und dort auch verarbeitet werden müssen. Je nach Größe der Dokumente können das zwischen 5 bis 100 Megabytes pro Dokument sein (im Durchschnitt sind es im Falle normaler Nachrichten nur 50-150 Kilobytes). ITyX empfiehlt deshalb das System in einem lokalen Netzwerk oder unter Einsatz einer Hochgeschwindigkeitsleitung zwischen Workstation und Mediatrix-Server zu betreiben (10Megabit/Sekunde, Up- und Download).

### 2.1 Datenbank

Für eine On-Premise-Lösung (System ist vor Ort installiert) werden folgende Datenbankmanagementsysteme unterstützt:

- MySQL ab v5.1 (erprobt bis v5.6)
- MSSQL ab 2005 (erprobt bis 2014)
- Oracle 11g (R2)

Neuere Versionen von Datenbanksystemen sollten vorher getestet werden, denn wenn sich die Verhaltensweisen verändert haben, kann der Einsatz von z.B. MSSQL 2014 zu Seiteneffekten führen, die vor dem Einsatz auf einer produktiven Umgebung behoben werden sollten. Grund dafür ist die Anpassung von Ausführungsplänen oder Indizes, die unter starken Belastungen besser auf die Datenbasis angepasst werden müssen.

Datenbanken, unabhängig welche Software hierfür verwendet wird, lagern Ihre Daten in der Regel auf der Festplatte und führen Indizes, sofern sie öfter verwendet werden, im Arbeitsspeicher. Die Anforderungen an die Software sind entscheidend dafür, wie viel Arbeitsspeicher der Datenbank zur Verfügung gestellt wird. Einer MySQL-Datenbank kann mit 2 Gigabyte RAM so gut wie immer arbeiten, selbst wenn die Datenbank schon eine Gesamtgröße von 30 Gigabyte und mehr hat. Nur werden hier sehr wahrscheinlich nicht genügend Daten indiziert und im Arbeitsspeicher vorgehalten. Die meisten Anfragen werden auf der Festplatte abgefragt, was ggf. weder CPU noch Arbeitsspeicher sonderlich belastet, aber dennoch eher langsam vonstatten geht, wenn die Anzahl der Anfragen höher wird als die Festplatte in der gewünschten Zeit leisten kann. In diesem Fall empfehlen wir entweder den Arbeitsspeicher zu erhöhen oder die Datenbank auf Solid-State-Disks (SSD) bzw. Hybrid-Festplatten zu speichern, da auf diesen Festplattentechnologien die Zugriffszeiten sehr gering sind.

---

*Hinweis zum Oracle 11g R2 Cluster-Betrieb: ITyX verwendet JDBC-Treiber, um auf Datenbanken zuzugreifen. Das Node-Management von Oracle-Cluster-Datenbanken ist für diese Technik nicht transparent. Erfolgt über das Cluster-Management ein Node-Switch (Verbindung wird von Server 1 auf Server 2 umgeleitet), so verlieren die Dienste ihre Verbindung, die erst nach einem Neustart der Dienste wiederhergestellt wird. Solange durch die ITyX-Dienste nicht explizit OCI-Treiber angesprochen werden können, empfehlen wir die Datenbanken stets auf einem Node bzw. Stand-Alone-Datenbank-Server zu betreiben.*

---

## 2.2 Hardware

Aufgrund der Architektur der Software, nämlich alle Daten absolut zentral zu lagern und zur Verfügung zu stellen, ist es notwendig eine performante Infrastruktur (Netzwerk, PC-Leistung für Mediatrix-Server und für die Datenbank) einzuplanen. Prinzipiell kann die Mediatrix inkl. Context auf einem Dual-Core-Rechner mit 1,4 GHz, einer 1 Gigabyte großen Festplattenpartition und 2 Gigabyte Arbeitsspeicher gestartet und betrieben werden. Sobald allerdings Automatisierungsaufgaben und mehr als 5 Benutzer sich auf dem System anmelden, werden die Anforderungen steigen, da ansonsten die Wartezeiten auf neue Dokumente höher werden.

Szenario 0: Grundvoraussetzung (Rechner soll Datenbank, Mediatrix und Context betreiben):

- Dual-Core-CPU
- 2 Gigabyte RAM
- 1 Gigabyte Festplatte
- 2-5 gleichzeitig angemeldete Benutzer
- Netzwerk: 5 Megabit, synchrone Verbindung

Die Grundvoraussetzung reicht aus, um ein Projekt mit bis zu 10 Teilprojekten in Mediatrix sowie 5 Benutzer gleichzeitig zu versorgen. Context übernimmt nur einfache Aufgaben wie eine Informationsextraktion mit 5-10 Elementen und pro Tag kommen maximal 150 neue Dokumente ins System (alles E-Mails, Anhang nicht größer als 2 Megabyte). Um eine Vorstellung davon zu bekommen, wie Sie Ihr System am besten dimensionieren, hier ein paar Szenarien. Für ein für Ihr Unternehmen zugeschnittenes Hardware-Sizing, sollten Sie eine Frage an [customercare@ityx.de](mailto:customercare@ityx.de) stellen.

Szenario 1: Empfohlen für Small Business (1 Rechner für alles, allerdings mit mehr Last):

- Quad-Core-CPU
- 12 Gigabyte RAM
- 4 Gigabyte Festplatte
- 10-25 gleichzeitig angemeldete Benutzer
- Netzwerk: 10 Megabit, synchrone Verbindung

Context lädt diverse Informationen aus der Datenbank in den Arbeitsspeicher, damit diese schnell verfügbar sind. Ist also ein Kundenindex mit 200.000 Kunden und 20 Attributen zu laden, wird hier mindestens 800-1000 Megabyte Speicher notwendig sein. Soll eine Extraktion erfolgen, die den



Kunden auf Basis der genannten Indizes identifiziert, sind insgesamt ca. 2 Gigabyte RAM notwendig. Sollen dann mehr als eine Extraktion parallel laufen, erhöht sich der Speicher entsprechend. Die Leistung von Mediatrix erhöht sich mit der Anzahl der gleichzeitig angemeldeten Benutzer sowie dem Aufkommen von Dokumenten pro Tag. Je mehr Dokumente gleichzeitig bearbeitet oder angezeigt werden, desto höher ist die Aktivität des Dienstes auf der Datenbank, was heißt, dass Mediatrix und auch die Datenbank mehr Speicher benötigen um die Daten hochverfügbar zu halten. Damit die Mitarbeiter nicht auf die Verarbeitung der Last warten müssen, ist eine schnellere Internetleitung sinnvoll. Wenn 1 Benutzer ein 1 Megabyte großes Dokument herunterlädt, sind 5 Megabit/Sek akzeptabel, wenn jedoch 10 Mitarbeiter das gleichzeitig machen, wird es langsam.

Die Einsatzzwecke von Mediatrix sind zudem vielseitig. Wird Mediatrix nur als E-Mail-Client verwendet, kommt man server-seitig mit 8 Gigabyte zurecht, client-seitig wird der Client wahrscheinlich nicht über die 720-Megabyte-Grenze gehen. Sobald allerdings serverseitig Faxe und Briefe durch eine OCR geschickt, klassifiziert, extrahiert und anderweitig aufbereitet werden und somit Context-Daten im RAM vorgehalten werden müssen, steigt der RAM-Bedarf entsprechend an. Die folgenden Szenarien **schätzen** beispielhaft den Hardwarebedarf ab, eine sichere Aussage bekommen Sie jedoch durch den Professional Service von ITyX, sobald das Projekt näher spezifiziert wird.

Szenario 2: Empfohlen für Small Business - inkl. Context (1 Rechner für alles)

- Quad-Core-CPU (inkl. Hyperthreading)
- 48 Gigabyte RAM
- 10 Gigabyte Festplatte
- 10-25 gleichzeitig angemeldete Benutzer
- Context Klassifizierung
- OCR-Dokumente (gescannte Briefe oder Faxe)
- Netzwerk: 10 Megabit, synchrone Verbindung

Wie in Szenario 1 erwähnt steigt der Speicherverbrauch serverseitig mit der Verwendung von Context-Diensten. Die errechneten KI-Daten werden bei Start eines Context-Servers in den Arbeitsspeicher geladen, um die Datenbank zu schonen. Wird eine Lernmenge von Texten für die intelligente Klassifizierung verwendet, so steigt der RAM-Verbrauch proportional zu den ermittelten KI-Ergebnissen. Sind 10000 Texte mit durchschnittlich 3 Seiten in der Lernmenge, so ist davon auszugehen, dass Context 300-600kb RAM pro Seite benötigt<sup>3</sup>, also zwischen 9-15 Gigabyte RAM. Wenn die Texte nicht bereits elektronisch vorliegen, müssen diese durch eine OCR-Software laufen (z.B. ABBYY Finereader). Zum einen benötigt die Software ebenfalls ca. 2-4 Gigabyte RAM, entscheidender ist jedoch hier die CPU. Wenn also das Verhältnis „Elektronische Dokumente: Scans“ zugunsten der Scans ausfällt, wird ein Quadcore nicht ausreichend sein, sobald mehr als 60 Dokumente pro Stunde verarbeitet werden sollen. Die Geschwindigkeit OCR/Seite liegt bei Verwendung von aktuellen Prozessoren bei bis zu 15 Sekunden pro Seite. Werden der OCR 2 CPU-Cores zugewiesen, so schafft die

<sup>3</sup> Diese Benennung ist abstrakt zu sehen, die notwendigen Daten im RAM hängen stark von der Konfiguration des Context Categorizers ab

OCR 240 Seiten im besten Fall, Overhead von Context und OCR-Software nicht eingerechnet). Bleiben bei einem Quadcore noch 2 Cores übrig um diese Texte zu klassifizieren und Mediatrix-Benutzer zu bedienen. Selbst mit Hyperthreading wäre dieser PC in diesem Szenario bei einer dauerhaften Verwendung der OCR immer stark belastet. Die Verarbeitung in Mediatrix dürfte hier langsam sein. Wenn sowohl OCR als auch Context als auch Mediatrix zu jeder Zeit verfügbar sein müssen, so ist der Rechner entweder mit mehr CPUs ausstatten oder die Software muss auf mehrere PCs / Server aufgeteilt werden.

Da sowohl Context als auch Mediatrix Dokumente und Konfigurationen in einer Datenbank speichern, ist eine Aufteilung auf mehreren PCs einfach durchführbar.

#### Virtualisierte Server-Umgebungen

Der Einsatz von Mediatrix-Context auf virtuellen Maschinen ist grundsätzlich möglich, aber die Verhaltensweisen in Sachen Performanz können unter diesen Bedingungen nicht mehr garantiert werden. Die nichttransparente Verwendung mehrerer Applikationen, verteilt auf diversen Guest-Systemen, kann sich sehr negativ auf Zugriffe auf der Datenbank auswirken. Des Weiteren führt der On-Demand-Zugriff auf CPU- und RAM-Ressourcen durch Mediatrix auf virtualisierten Umgebungen zu schwer nachvollziehbaren Performanzproblemen und Seiteneffekten.

---

*Derzeit setzen wir aus den oben genannten Gründen immer physikalische Rechner als Basis für den Einsatz von Mediatrix-Context und der Datenbank voraus.*

---

### 2.3 Client

Die Anforderungen an den Client sind unterschiedlich. Es handelt sich um eine Java-Applikation, die auf jedem Betriebssystem gestartet werden kann, auf dem Java installiert ist. Wird Mediatrix nur für die Verarbeitung von Mails verwendet, so ist je nach Aufbau der Mails zwischen 200-600 Megabyte RAM notwendig (nur für die Applikation, der PC sollte entsprechend minimal 2 Gigabytes RAM haben). Wird Mediatrix jedoch als Context-Erweiterung, also als „Nachbearbeitungsstation“ verwendet, so sind auch bis zu 1,5 Gigabytes RAM notwendig, da die Dokumente je nach Größe und Aufbereitung mehr Ressourcen benötigen. Wir empfehlen für den Einsatz von Nachbearbeitungsstationen deshalb immer ein 64bit-Betriebssystem und entsprechend auch ein 64bit-Java.

---

*32bit-Betriebssysteme können maximal 3,2 Gigabytes Arbeitsspeicher verwalten, 32bit-Java-VMs (Mediatrix Client wird in eine JVM (Java Virtual Machine) gestartet) können sogar nur ca. 1,1 Gigabytes RAM zuweisen. 64bit-Varianten haben diese Begrenzung nicht (sie liegt bei 16 Exabytes bei Betriebssystemen<sup>4</sup> und Java).*

---

Bei der Verarbeitung von Dokumenten mit Mediatrix ist auch die Bandbreite zwischen Server und Client immer zu berücksichtigen. Da die Daten nie auf der Workstation vorgehalten werden (Ausnahme

---

<sup>4</sup> Microsoft Windows hat hier eine geringere Grenze, bei Server-Betriebssystemen liegt diese bei 192 Gigabytes

ist hier der Operator-Modus, muss ein Dokument im Ganzen vom Server zum Client gesendet und nach der Bearbeitung wieder zurück gesendet werden. Wenn Mediatrix als Nachbearbeitungsstation verwendet wird, so kann ein Dokument mehrere Megabytes groß sein. Ist die Bandbreite nicht ausreichend, vor allem wenn mehrere Benutzer über dieselbe Leitung arbeiten, so wird als Alternative häufig eine Terminal-Server-Lösung vorgeschlagen. Die Clients werden auf einem Remote-Server gestartet und über eine entsprechende Software (Cisco, Remote Desktop, SSH) verwendet. Der Terminal-Server sollte entsprechend „nahe“ beim Mediatrix-Server stehen, um das Bandbreite-Problem zu lösen. Für jeden aktiven Benutzer muss dieser Terminal-Server allerdings ausreichend Arbeitsspeicher für die aktive Session des Betriebssystems und den zu verwendenden Mediatrix-Client bereitstellen (also ca. 2-4 Gigabytes RAM pro Benutzer). Terminal-Server-Lösungen sind dann empfehlenswert, wenn die Bandbreite für die Mitarbeiter beschränkt ist.

Beispiel:

Ein Dokument hat aufgrund der Auflösung eine durchschnittliche Größe von 5 Megabytes.

Die Bandbreite ist synchron und full-duplex 2 Megabit/Sekunde, also 256 Kilobyte/Sekunde. Das bedeutet, der Down- und Upload dauert für ein solches Dokument jeweils 20 Sekunden. Im Operator-Modus werden standardmäßig 5 Dokumente vorgeladen, die im Arbeitsspeicher des Clients verweilen. Alle 20 Sekunden ist nach Start des Clients also 1 Dokument verfügbar, sofern die Bandbreite immer zur Verfügung steht. Fängt der Benutzer an diese Dokumente zu verarbeiten, so ist die Verarbeitungsgeschwindigkeit genau dann ausreichend, wenn die Verarbeitung eines Dokuments 15-25 Sekunden dauert. Würde der Benutzer nur 5 Sekunden benötigen, dann wären Wartezeiten (Ladezeiten) einzubeziehen. Dieses Problem ist also nur mit höherer Bandbreite lösen. Ist diese am derzeitigen Standort nicht verfügbar, so kommt eine Terminal-Lösung in Frage.

Mediatrix kann auch mit weniger Bandbreite, genauer gesagt mit minimalen 0,3 Mbit pro Benutzer betrieben werden. Dies schränkt jedoch die Performanz genau dann stark ein, wenn regelmäßig größere Dokumente bearbeitet werden müssen. Insbesondere sind Updates, die über Java-Webstart an die Workstations gesendet werden, langsam. In manchen Troubleshooting-Fällen wird empfohlen auf den Workstation-PCs wird empfohlen die Client-Software neu zu installieren (Stichwort: `javaws -uninstall`). Sei es aus Troubleshooting- oder Update-Gründen: Das erneute Herunterladen der Software beeinträchtigt bei niedriger Bandbreite zum einen die Arbeitszeit des Agenten, der auf den Start von Mediatrix warten muss, zum anderen kann der Download auch die Performanz anderer Agenten, die mit derselben Netzwerkleitung arbeiten, negativ beeinflussen. Die Reaktionszeit zwischen Client und Server wird bei unterschiedlich großen Dokumenten, die bearbeitet und zurückgesendet werden, ggf. sehr langsam oder zumindest ungleichmäßig sein.

Technisch ist es möglich Mediatrix mit wenig Bandbreite zu betreiben, es muss allerdings mit diesen Problemen gerechnet werden. Die Lösungen sind jeweils individuell zu erstellen, z.B. beim Download des Programms einen Proxy-Server oder http-Cache ansprechen, der die Java-Dateien nur 1x herunter lädt und dann allen Clients zur Verfügung stellt.

## 2.4 Speicher

*Die Zuweisung von Speicher erfolgt über die Konfigurationsdateien der jeweiligen Dienste.*

Die Server-Dienste von Mediatrix (Mediatrix WebApp, Emailedemon, Outbounddemon etc.) benötigen in der Regel nicht viel Speicher. Standardmäßig werden jedem Dienst maximal 1,4 Gigabytes Arbeitsspeicher zugewiesen. Der Bedarf der Dienste ist auch bei höherem Leistungsbedarf (mehr als 40 Benutzer gleichzeitig aktiv) in der Regel nicht viel höher, jedoch wird jedes Ergebnis einer Datenbank-Aktion temporär im Speicher des jeweiligen Dienstes vorgehalten. Es wird der Java-Umgebung überlassen, wann dieser Speicher wieder freigegeben wird. In der Regel erfolgt dies automatisch, sobald die Leistungsforderungen kurzfristig geringer ausfallen. Das Stichwort ist hier „Garbage Collection“, dieser Vorgang entfernt ungenutzte Datenfragmente aus dem Speicher. Dieser Vorgang kostet jedoch CPU-Zeit, was ggf. zu Wartezeiten bei den Benutzern führen kann. D.h. je nach Benutzer-Anzahl kann es leistungstechnisch besser sein den Arbeitsspeicher auf höhere Werte zu setzen, damit möglichst wenig Zeit für Garbage Collection aufgewendet werden muss. Für eine „Faustregel“ können 200-400 Megabyte RAM pro Benutzer angenommen werden. 10 Benutzer werden serverseitig mit maximal 4 Gigabyte versorgt sein, 50 Benutzer könnten bis zu 20 Gigabytes benötigen. Die maximale Auslastung wird jedoch nur dann erreicht, wenn alle Benutzer dauerhaft Datenbank-Anfragen erzeugen. Es ist jedoch zu erwarten, dass ein Mensch z.B. auch mal ein Dokument lesen muss, sodass sich die Reservierung des Arbeitsspeichers sehr wahrscheinlich bei ca. 8 Gigabytes einpendeln wird.

Der Speicherverbrauch der Subdienste ist nicht von der Anzahl der Benutzer, sondern von ihren Aufgaben abhängig. Kommen 2 E-Mails pro Minute an, so wird der Dienst „Emailedemon“ nicht mehr als 120 Megabytes RAM benötigen. Sind jedoch 400 Mails pro Minute von 20 verschiedenen Konten zu erwarten, so ist es möglich, dass das Programm für eine performante Abarbeitung 3-5 Gigabyte Arbeitsspeicher benötigt. Größen, die ebenfalls zu berücksichtigen sind, wären in diesem Falle auch das Abhol-Intervall (jede Minute? alle fünf Minuten?) sowie die durchschnittliche Größe der E-Mails.

Beim Dienst „Escalationdemon“ ist es ähnlich. Für Teilprojekte sind Eskalationszeiten zu konfigurieren (Kapitel [Fehler! Verweisquelle konnte nicht gefunden werden.](#)). Je nach Häufigkeit der Eskalation wird ggf. mehr Arbeitsspeicher benötigt, um die gestellten Aufgaben (Eskalationsmarkierungen) zu leisten, in der Regel wird der Standardwert von 1,4 Gigabytes RAM selten verwendet. Eine „Faustregel“ für „Maxmemory“ ist beim Escalationdemon grob gerechnet (Anzahl zu eskalierender Dokumente pro Minute \* 5 Megabyte RAM).

Der Speicherverbrauch des Dienstes „PTME“ ist nicht konstant. Der Dienst bezieht die KI-Informationen aus der Datenbank. Je nachdem wie viele Informationen hier gespeichert sind, steigt auch der Speicherbedarf. Wird die PTME aktiviert, so werden Benutzer durch ihre tägliche Arbeit KI-Daten erzeugen und somit den notwendigen Speicher des PTME-Dienstes in die Höhe treiben. Zu Beginn ist mit einem Verbrauch von 300 Megabytes RAM zu rechnen, sobald jedoch z.B. 10 Millionen Einträge in der SQL-Tabelle `txtsems` enthalten sind, wird die PTME zwischen 5-8 Gigabyte RAM

benötigen. Die KI-Daten beinhalten Textbausteinvorschläge und Inhaltsanalysen zur automatischen Zuweisung an Mediatrix-Teilprojekte.

Die Hardware-Anforderungen von Contex sind ebenfalls nicht konstant, da der RAM-Verbrauch davon abhängt, welche Dokumente zu analysieren sind. „Contex Server“ speichert die Klassifizierungsalgorithmen (Categorizer, Rulebased Categorizer etc.), d.h. auch einen Teil der verwendeten Lernmenge, sowie die Extraktions-Strukturen (Information Extraction) im RAM. Die Teil-Ergebnisse der Algorithmen werden ebenfalls kurzzeitig im Speicher belassen, bevor das End-Ergebnis ermittelt werden kann. Je nach Größe des Dokuments können diese Elemente den Dienst „Contex Server“ anwachsen lassen. Meistens reichen 2 Gigabyte RAM aus, ab einer Dokumentengröße von 20 Seiten oder einer Categorizer-Lernmenge von 4000 Dokumenten können es aber auch 4-6 Gigabyte RAM werden.

Der Speicherverbrauch des Contex Schedulers, über den die Designer-Prozesse gestartet werden, ist abhängig von den Prozessen und der Anzahl der gleichzeitig laufenden Threads. In der Regel ist der Verbrauch niedrig (1 Gigabyte RAM), wenn jedoch Programmierung in Form von Java-Beans in die Prozesse mit eingebaut wird, kann der Verbrauch immens steigen – Es kann nicht pauschal abgeschätzt werden ob der Mehrverbrauch nur wenige Megabytes ausmacht oder sich verachtfacht.

Mediatrix bietet zahlreiche Möglichkeiten um während des Arbeitsablauf „Business Rules“, also kundenspezifischen Code, einzufügen. Der Speicherverbrauch kann nur während des Projekts ermessen werden, muss allerdings beim Hardware-Sizing frühzeitig berücksichtigt und individuell geschätzt werden. Beispielszenario; beim Abholen von E-Mails (RAM-Verbrauch geht zulasten des Dienstes „Emaildaemon“):

- Nach jeder erfolgreich abgeholten E-Mail ist ein spezieller Log-Eintrag in eine Datei erforderlich → geringer RAM-Verbrauch
- Nach jeder erfolgreich abgeholten E-Mail ist jeder Bild-Anhang über eine OCR zu erkennen und als Text an den E-Mail-Body zu knüpfen → hoher RAM-Verbrauch, insbesondere auch Rechenaufwand

## 3 Grundinstallation des Systems

### 3.1 Datenbank-Setup

Erstellen Sie eine neue Datenbank und, falls gewünscht, einen separaten Benutzer für diese Datenbank. Z.B. unter MySQL

- `mysql -u Datenbankbenutzer -p`
- `create database mediatrixDB;`
- `create database ContextDB;`

Falls kein administrativer Benutzer erwünscht ist, stellen Sie sicher, dass der Datenbank-Benutzer die Rechte `INSERT`, `UPDATE`, `SELECT`, `ALTER` und `CREATE TABLE`, sowie die Zugriffsrechte auf die Tabellen selbst, besitzt.

---

*Stellen Sie sicher, dass genau diese Benutzer und Datenbanknamen in den Dateien `mediatrix.properties` und `context.properties` eingetragen sind.*

---

#### Hinweise zu MySQL

Im Installationsverzeichnis des MySQL Server die Datei `my.ini` öffnen und den Eintrag `max_allowed_packet=1G` hinzufügen. Standardmäßig sind 2 Megabyte eingerichtet, was für größere Dokumente zu wenig ist.

---

*Ab MySQL 5.6 liegt diese Datei in `%programdata%/MySQL/MySQL Server 5.x`*

---

### 3.2 Installationsverzeichnis anlegen

Kopieren Sie das Archiv (z.B. `mediatrix-2.5.4.tar`) auf Ihren Server und unpacken sie es in ein entsprechendes Verzeichnis, z.B.

- `/opt/mediatrix/`
- `D:\mediatrix\`

Wenn Sie die Datenstruktur der Installation gezielt kontrollieren möchten, schränken Sie diese auf folgende Rechte ein, bezogen auf den Benutzer, der die Programme ausführen soll:

Unterverzeichnis	Linux	Windows
<b>Activemq/</b>	500	Lesen
<b>Activemq/data</b>	600	Lesen + Schreiben
<b>Bin/</b>	500	Lesen

Unterverzeichnis	Linux	Windows
<b>Bin/service/data</b>	600	Lesen + Schreiben
<b>Conf/</b>	400	Lesen
<b>Data/</b>	600	Lesen + Schreiben
<b>Es/</b>	400	Lesen
<b>Es/bin</b>	500	Lesen
<b>Es/data</b>	600	Lesen + Schreiben
<b>Es/logs</b>	600	Lesen + Schreiben
<b>Init-scripts/</b>	500	Lesen
<b>Libs/</b>	400	Lesen
<b>Licenses/</b>	400	Lesen
<b>License-service</b>	500	Lesen
<b>Logs/</b>	600	Lesen + Schreiben
<b>tmp/</b>	600	Lesen + Schreiben
<b>Tomcat/</b>	700	Lesen + Schreiben

Damit Mediatrix-Contex mit der Datenbank kommunizieren kann, ist eine Treiber-Datei zu installieren. Es ist lediglich der Treiber notwendig, der verwendet werden soll. Sie müssen also nicht alle Treiber-Dateien laden.

---

*ITyX liefert aus Lizenzgründen standardmäßig keinen Datenbanktreiber mit; diese müssen vom Kunden selbst heruntergeladen werden. Die Treiber stehen kostenlos zur Verfügung. Mediatrix-Contex wird ohne den Treiber nicht starten können. Eine Liste mit URLs, wo diese Treiber zu finden sind, finden Sie unter dieser Box.*

---

Mediatrix-Contex wird das Verzeichnis `mediatrix/libs/` gänzlich nach JAR-Dateien durchsuchen. Der Datenbanktreiber, in der Regel `ojdbc.jar`, `sqljdbc.jar` oder `mysql-connector.jar`, sollte der Übersicht halber unter `mediatrix/libs/db/` gespeichert werden.

### MySQL

URL <http://www.mysql.de/products/connector/>

Instruktionen Suchen Sie den Link „JDBC Driver for MySQL (Connector/J)“ und laden Sie den Treiber herunter.

#### Microsoft SQL Server

URL <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=21599>

Instruktionen Suchen Sie die entsprechende Datei aus und laden Sie diese herunter. Im Archiv sollten jeweils eine sqljdbc.jar (MSSQL 2005) sowie eine sqljdbc4.jar (MSSQL 2008 oder höher) enthalten sein.

#### Oracle

URL <http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>

Instruktionen Suchen Sie die entsprechende Version des Treibers herunter. Um Dateien bei <http://www.oracle.com> herunterzuladen müssen Sie sich einmalig registrieren.

### 3.3 Konfigurationsdateien anpassen

Die server-seitigen Konfigurationsdateien befinden sich unter `mediatrix-Context/conf/`. In diesen Dateien werden globale Einstellungen getroffen. Für den Anfang reichen die vorbereiteten Versionen, die mit ausgeliefert werden.

---

*Es müssen mindestens die Datenbank-Konfigurationen für Contex und Mediatix angepasst werden, um die Installation erfolgreich abzuschließen. Auch wenn Contex nicht genutzt wird, ist eine (leere) Datenbank notwendig, da diese auch vom Lizenz-System verwendet wird.*

---

Icon	Beschreibung
mediatrix.properties	Serverseitige Einstellungen für Mediatix-AppServer, -Email demon, -Escalation demon, -Outbound und -ReportScheduler
context.properties	Serverseitige Einstellungen für Contex Services (Contex Server, Scheduler)

#### Datei: mediatrix.properties

```
dbappendix=mysql5
dbhost=127.0.0.1
dbname=mediatrix20
dbuser=root
dbpassword=ityx
dbport=3306
```



dbappendix	Über diese Einstellung wird definiert, welches Datenbankmanagementsystem verwendet wird. Neben „mssql“, „mysql5“, „mysql_unicode“ und „oracle“ sind auch diverse vordefinierte Varianten möglich, die auf andere Systemkonfigurationen zugeschnitten sind. Siehe Kapitel <a href="#">4.4.1 (Datenbank-Einstellungen, Seite 46)</a> .
dbhost	DNS oder IP-Adresse des Servers, auf dem die Datenbank läuft.
dbname	DB-Username.
dbpassword	DB-Password.
dbport	Port der Datenbank(-Instanz). Der Standardport für MySQL ist 3306, Oracle 1521 und MSSQL 1433

*Datei: contex.properties*

```
host=127.0.0.1
database=Contex20
login=DB-Username
password=DB-Password
port=3306
dbappendix=mysql5
```

dbappendix	Über diese Einstellung wird definiert, welches Datenbankmanagementsystem verwendet wird. Neben „mssql“, „mysql“, „mysql_unicode“ und „oracle“ sind auch diverse vordefinierte Varianten möglich, die auf andere Systemkonfigurationen zugeschnitten sind.
host	DNS oder IP-Adresse des Servers, auf dem die Datenbank läuft.
name	DB-Username.
password	DB-Password.
port	Port der Datenbank(-Instanz). Der Standardport für MySQL ist 3306, Oracle 1521 und MSSQL 1433, 1434

---

*Die Einstellungen sind zwischen den beiden Dateien nicht zu kopieren, weil die Einträge leicht unterschiedlich sind.*

---

### 3.4 Datenbanken initialisieren

Schließen Sie die Installation der Software mit dem Anlegen der Datenbank ab. Dabei ist es wichtig sowohl die Contex- als auch die Mediatrix-Datenbank zu initialisieren, selbst wenn Sie nur eines der beiden Produkte nutzen. Mit der Initialisierung werden Datenbank-Tabellen sowie eine Minimal-Konfiguration des Systems innerhalb der Datenbank durchgeführt.

Script	Beschreibung
<b>Mediatrix/bin/install/04_mediatrixupdate.*</b>	Initialisierung bzw. Update der Mediatrix-Datenbank-Struktur
<b>Mediatrix/bin/install/01_contextDBinit.*</b>	Initialisierung bzw. Update der Contex-Datenbank-Struktur

Dieselben Scripts werden auch für Updates eines bestehenden Systems verwendet. Sie setzen die bestehende Datenbank-Konfiguration nicht zurück, weshalb sie theoretisch beliebig oft gestartet werden können, insbesondere in gleichen Minor-Versionen (z.B. 2.5.4 → 2.5.5). Wenn Updates mit größeren Versions-Sprüngen durchgeführt werden sollen, z.B. 2.2 → 2.5, ist das entsprechende Kapitel im Migrationshandbuch zu beachten, bevor das Update durchgeführt wird.

### 3.5 Registrierung der Software

Mediatrix-Contex unterliegen einem kontingent-begrenztem Lizenzierungsmodell. Damit die Software betrieben werden kann sind folgende Lizenzkomponenten notwendig:

Master-Lizenz	Enthält die Lizenznehmer-Daten, die bei Vertragsabschluss mit ITyX erfasst wurden. Sie bestimmt unter anderem die Dauer der Lizenzgültigkeit und wie viele Contex-Master verwendet werden dürfen.
Runtime-Lizenz	Die Laufzeit-Lizenz, die die vordefinierten Mediatrix-Contex-Software-Module („welche Contex-Features dürfen verwendet werden?“) und das Kontingent definiert.

---

*Kopieren Sie die Master-Lizenz-Datei in das Verzeichnis `/mediatrix-Contex/conf/`*

---

Das Lizenz-Manager-Tool muss einmalig gestartet werden. Dieses befindet sich unter `mediatrix-Contex/bin/00_licensemanager`). Für die Ausführung des Programms ist Windows oder ein XServer (Linux) notwendig. Wenn die Master-Lizenz erfolgreich erkannt wurde, wird sich folgendes Fenster öffnen ([Abbildung 1](#)).

Licensee: ITyX India

Additional Information: systemadmin domain

Guarantor: [REDACTED]

Address: Carl-Benz-Strasse 10-12, Mülheim-Kärlich, Mülheim-Kärlich, 56218 [REDACTED]@ityx.de

Domain: admin domain

Projects: 100

EOT: 31 May, 2017 12:00:00 AM

(If there is a problem while loading a license it is turned into "locally disabled".)

If you manually disable/enable a license locally, please restart all servers, don't reload the license tracker! This is due to system safety!

Refresh

Reload License Tracker

Active	Default	Token	File	Volume	Used	Type	per Document	per Year	EOT	Funcio...	Locally disabled

Add Runtime License   Create Activation File   Load Activated Runtime   Create Deactivation File   Load Deactivated Runtime   Set Default License

Abbildung 1: License Manager

**Abbildung 1** zeigt den License Manager. Die Daten über Lizenznehmer und Adressdaten werden aus der Master-Lizenz geladen. Ohne eine solche Lizenz würde das Programm nicht starten. Um Mediatrix und Context zu nutzen ist zudem eine Runtime Lizenz notwendig.

#### Kurz-Anleitung

- Wählen Sie eine Laufzeit-Lizenz (runtime.\*) aus und bestätigen Sie den Auswahldialog
- Klicken Sie auf „Create Activation File“
- Lassen Sie die erstellte Datei durch ITyX aktivieren. Wenden Sie sich an Ihren Projektleiter oder nutzen Sie die Webseite [https://license.ityx.de:8443/license\\_activation\\_client/](https://license.ityx.de:8443/license_activation_client/)
- Klicken Sie auf „Load Activated Runtime“ und wählen Sie die Ihnen zugesendete Datei aus
- Vergeben Sie eine Bezeichnung, die als Identifikationsmerkmal Ihrer Lizenz dient (beliebiger Text)
- Definieren Sie die systemweite Standard-Lizenz

#### Detail-Anleitung

- Wählen Sie eine Laufzeit-Lizenz (runtime.\*) aus und bestätigen Sie den Auswahldialog (**Abbildung 2**)

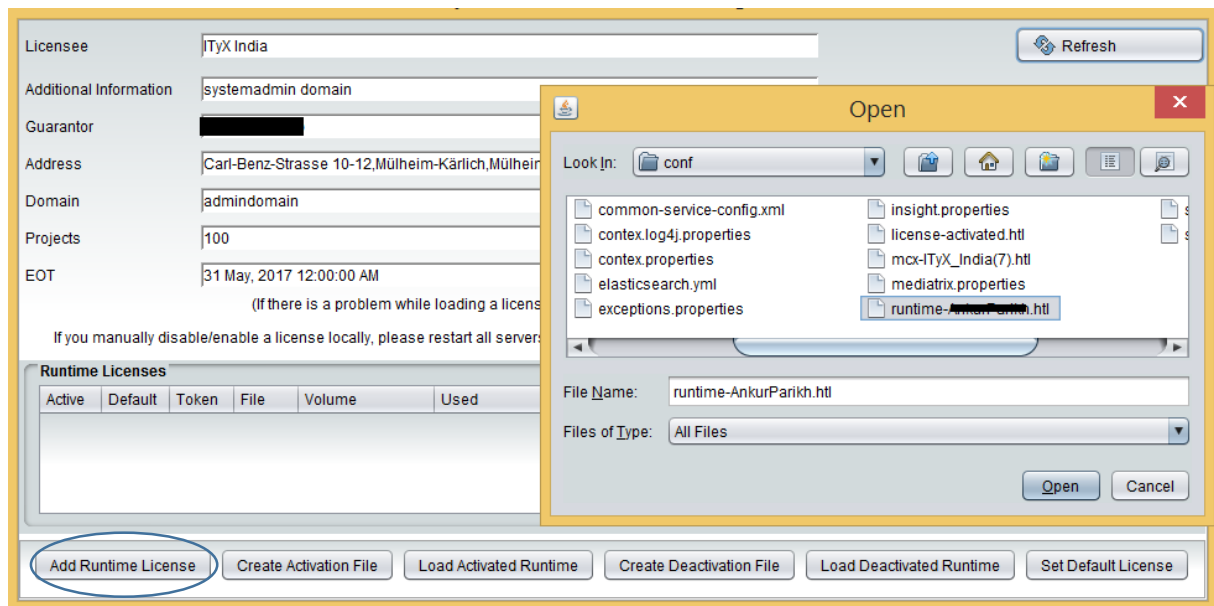


Abbildung 2: License Manager - Add Runtime License

- Klicken Sie auf „Create Activation File“ um eine Aktivierungsdatei der Form „activation-runtime-`<yourname>.htl.dat`“ zu generieren (Abbildung 3)

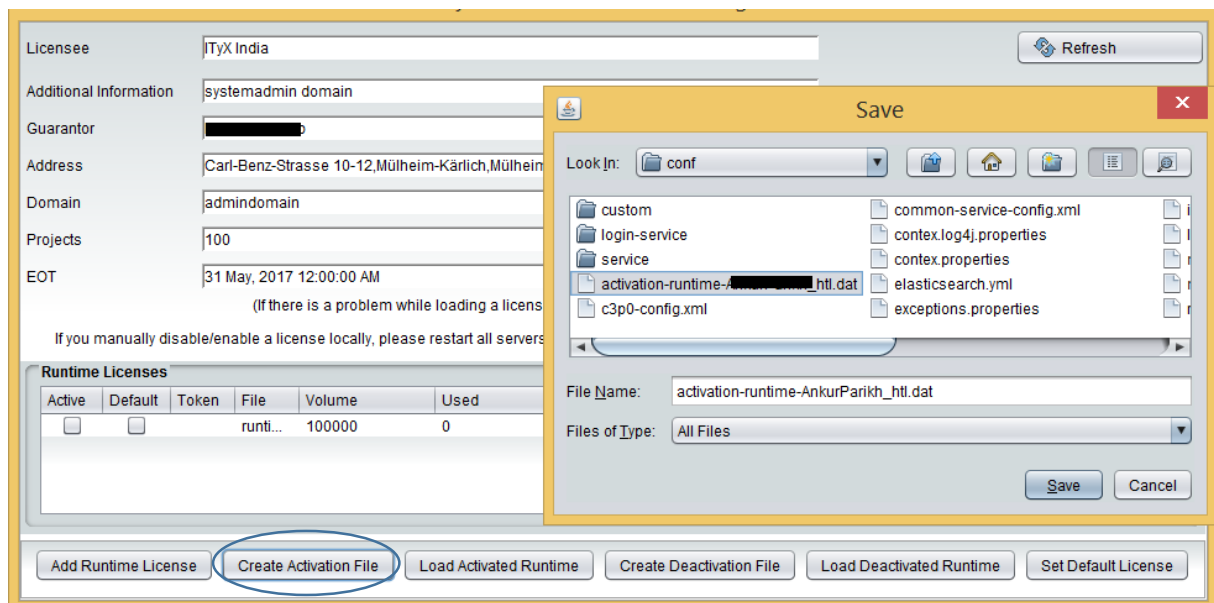


Abbildung 3: License Manager - Create Activation File

- Speichern Sie die erstellte DAT-Datei unter mediatrix\conf (wie alle anderen Lizenz-Dateien auch)
- Lassen Sie die erstellte Datei durch ITyX aktivieren. Wenden Sie sich an Ihren Projektleiter oder nutzen Sie die Webseite [https://license.ityx.de:8443/license\\_activation\\_client/](https://license.ityx.de:8443/license_activation_client/)

- Die erstellte Datei hat in der Regel einen Namen der Form „activated-runtime-<yourname>\_htl.dat“. Speichern Sie diese unter mediatrix\conf

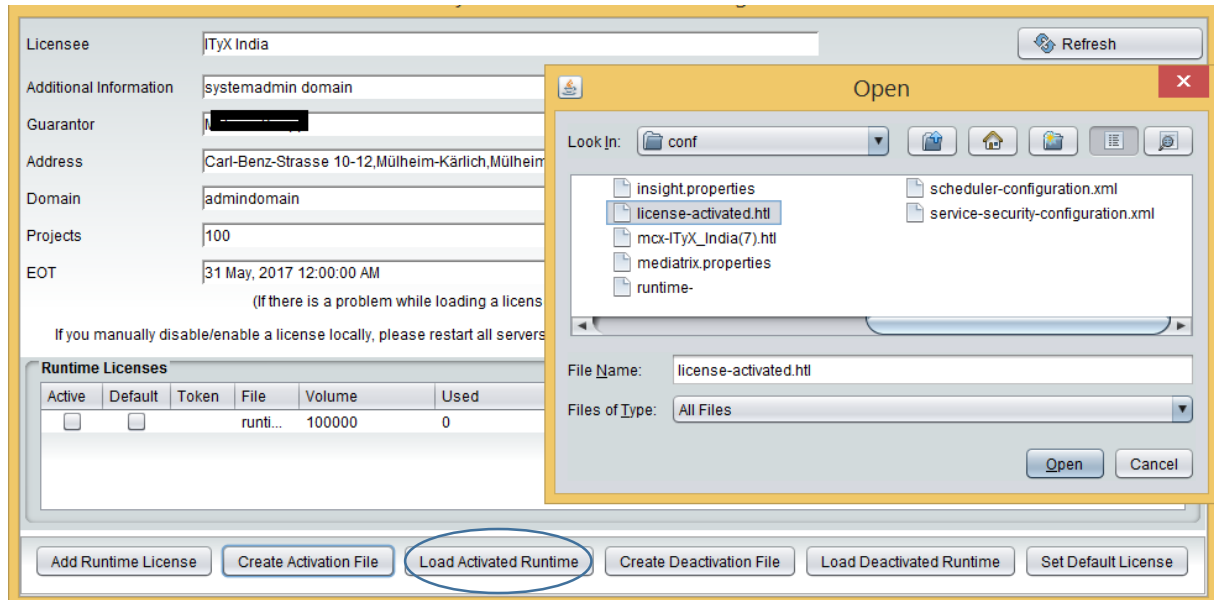


Abbildung 4: License Manager - Load Activated Runtime

- Klicken Sie auf „Load Activated Runtime“ und wählen Sie die Ihnen zugesendete Datei aus (Abbildung 4)

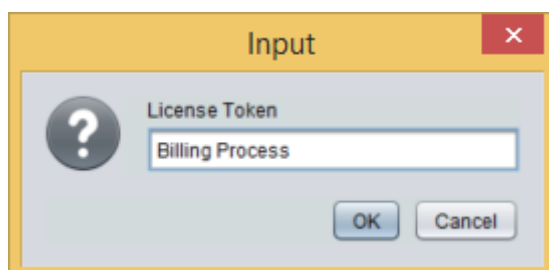


Abbildung 5: Vergabe des License Tokens (Lizenz-Bezeichnung)

- Vergeben Sie eine Bezeichnung, die als Identifikationsmerkmal Ihrer Lizenz dient (beliebiger Text, Abbildung 5). Über diese Bezeichnung werden Dokumente durch die ITyX-Systeme wie Mediatrix und Context der Lizenz zugeordnet und registriert
- Bestätigen Sie den Dialog, dass die gewählte Lizenz-Bezeichnung korrekt ist (Abbildung 6)

*Hinweis: Es können mehr als eine Runtime-Lizenz mit unterschiedlichen Nutzungsrechten und Laufzeiten installiert werden, über verschiedene eindeutige License Tokens kann der Verbrauch jeder Lizenz zu einem bestimmten Konto oder Prozess definiert werden.*

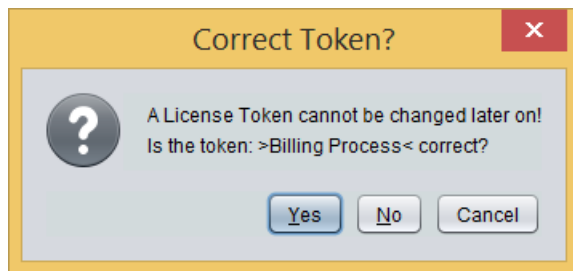


Abbildung 6: License Manager - Correct Token?

- Wählen Sie einen der Einträge im Bereich „Runtime Licenses“ und nutzen Sie den Button „Set Default License“ (Abbildung 7), um zu definieren, unter welcher Runtime-Lizenz Seiten konsumiert werden sollen, wenn keine Lizenz angegeben wurde. Lizenzen können in den folgenden Modulen definiert werden
  - Context Scheduler
  - Mediatrix Projektadministration
  - Mediatrix Konten

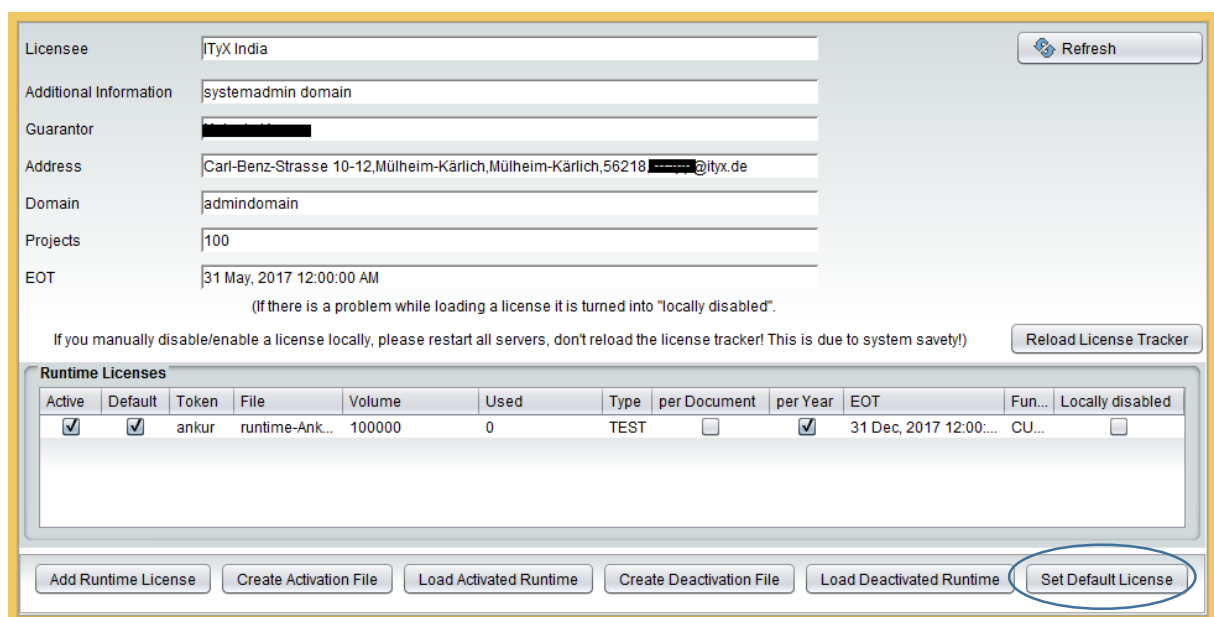


Abbildung 7: License Manager - Set Default License

*Hinweis: OCR ist eine 3rd-Party-Komponente, die verwendeten OCR-Seiten werden nicht zusätzlich von der Lizenz abgezogen. Wenn die OCR-Software ebenfalls volumenbasiert arbeitet (z.B. Abbyy Finereader OCR), so wird in der Regel von ITyX eine separate Lizenz für diese Software verkauft. Der OCR-Verbrauch wird jedoch nicht über den ITyX-License Manager widerspiegelt.*

*Installation unter Linux:*

Für den Start des License-Managers ist ein X-Server notwendig. Das Tool muss nicht zwingend auf dem Rechner gestartet werden, auf dem der Dienst später laufen soll. Sie können dieses auch über SSH-X oder auf einem Windows-Rechner starten, indem Sie das Mediatrix-Verzeichnis dorthin kopieren. Entscheidend ist die Datenbank-Verbindung, und dass der License-Service als Dienst auf dem Rechner läuft, auf dem er später auch eingesetzt wird. Die Registrierung erfolgt über eine Bindung der Lizenz an die Hardware des Rechners, weshalb Ihnen die Aktivierung auf einem anderen Rechner ggf. Probleme bereiten könnte.

### 3.6 Dienste installieren

#### 3.6.1 Windows Dienste installieren

Um die Programme als Windows-Dienst zu etablieren ist im Verzeichnis `c:\mediatrix\bin\install` die Datei `00_installService.cmd` zu starten. Um Dienste zu installieren sind Administrator-Rechte notwendig.

- Im Dienste-Fenster (Verwalten → Service) zuerst den „ITyX License Service“, anschließend „ITyX IDig Indexserver“, „ITyX Contex Server“ und danach die anderen Dienste starten

---

*Nach der erfolgreichen Konfiguration aller Mediatrix-Dienste empfiehlt es sich den „ITyX AppServer“ erst als letzten Dienst zu starten, da je nach Konfiguration Module von anderen Diensten benötigt werden. Dazu gehören Escalation-Demon, PTME. Contex Server muss ebenfalls vor dem ITyX AppServer online sein.*

---

Die Konfigurationsdateien aller installierbaren Dienste befinden sich im Verzeichnis `conf/service`. Die Änderungen innerhalb dieser Dateien sind erst nach einem Neustart des jeweiligen Dienstes wirksam.

---

*Sollten mehrere Java-Versionen auf dem Rechner installiert sein, muss die zu nutzende Umgebung in der Datei `ityxcommon.inc.conf` eingetragen werden, z.B.:*

*# Java Executable<sup>5</sup>*

***wrapper.java.command=%JAVA\_HOME%/bin/java<sup>6</sup>***

---

In diesen Dateien können Speichergrenzen und Systemeinstellungen gesetzt werden. Als Startparameter sollten die Standardeinstellungen von maximal 1400MB pro Dienst ausreichen.

---

<sup>5</sup> Mit „#“ beginnende Zeilen sind auskommentiert

<sup>6</sup> Die Umgebungsvariable `JAVA_HOME` ist, falls nicht schon geschehen, auf Betriebssystem-Ebene einzustellen, so dass sie auf das BIN-Verzeichnis von Java zeigt, z.B. „`c:\Program Files\Java\jdk1.8.0_60`“

### 3.6.2 Linux Dienste installieren

Die Installation der Dienste unterscheidet sich zur Prozedur unter Windows. Es wurden Bourne-Bash-Shell-Scripts generiert, die mit jeder Installation von Mediatrix-Contex ausgeliefert werden. Sie befinden sich unter:

```
mediatrix/init-scripts/
```

Alle Scripts müssen ausführbar sein, um gestartet zu werden zu können (`chmod a+x init-scripts/*`). Registrieren Sie diese Scripts in geeigneten Runlevel Ihres Linux-Servers, um diese als Daemon verfügbar zu halten. Stellen Sie sicher, dass die Startreihenfolge berücksichtigt wird. Es folgt ein Beispiel den Auszug einer Bash-Datei, um ITyX-Dienste als Linux-Daemon zu laden.

```
#!/bin/sh
baseDir=/opt/mediatrix
pidDir=/var/run
#
# Copyright (c) 1999, 2009 Tanuki Software, Ltd.
# http://www.tanukisoftware.com
# All rights reserved.
#
# This software is the proprietary information of Tanuki Software.
# You shall use it only in accordance with the terms of the
# license agreement you entered into with Tanuki Software.
# http://wrapper.tanukisoftware.org/doc/english/licenseOverview.html
#
# Java Service Wrapper sh script. Suitable for starting and stopping
# wrapped Java applications on UNIX platforms.
#
#-----
# These settings can be modified to fit the needs of your application
# Optimized for use with version 3.3.5 of the Wrapper.
#User who starts this application
#USER=meister
# Application
APP_NAME="idigserver"
APP_LONG_NAME="Ityx-Contex-Server"
Seite 250 von 370
Mediatrix-Contex: Administration Linux
# Wrapper
WRAPPER_CMD=$baseDir"/bin/service/linux64/wrapper"
WRAPPER_CONF=$baseDir"/conf/service/ityx-Contex-server.conf"
# Priority at which to run the wrapper. See "man nice" for valid priorities.
# nice is only used if a priority is specified.
```



```

PRIORITY=
# Location of the pid file.
PIDDIR=$pidDir
# If uncommented, causes the Wrapper to be shutdown using an anchor file.
# When launched with the 'start' command, it will also ignore all INT and
# TERM signals.
#IGNORE_SIGNALS=true
# Wrapper will start the JVM asynchronously. Your application may have some
# initialization tasks and it may be desirable to wait a few seconds
# before returning. For example, to delay the invocation of following
# startup scripts. Setting WAIT_AFTER_STARTUP to a positive number will
# cause the start command to delay for the indicated period of time
# (in seconds).
#
WAIT_AFTER_STARTUP=0
# If set, the status, start_msg and stop_msg commands will print out detailed
# state information on the Wrapper and Java processes.
#DETAIL_STATUS=true
# If specified, the Wrapper will be run as the specified user.
# IMPORTANT - Make sure that the user has the required privileges to write
# the PID file and wrapper.log files. Failure to be able to write the log
# file will cause the Wrapper to exit without any way to write out an error
# message.
# NOTE - This will set the user which is used to run the Wrapper as well as
# the JVM and is not useful in situations where a privileged resource or
# port needs to be allocated prior to the user being changed.
#RUN_AS_USER=
# The following two lines are used by the chkconfig command. Change as is
# appropriate for your application. They should remain commented.
# chkconfig: 2345 20 80
Seite 251 von 370
Mediatix-Contex: Administration Mediatix- und Contex Installer
# description: ityxContexserver
# Initialization block for the install_initd and remove_initd scripts used by
# SUSE linux distributions.
### BEGIN INIT INFO
# Provides: ityxContexserver
# Required-Start: $local_fs $network $syslog
# Should-Start:
# Required-Stop:
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: ityxContexserver

```

```
# Description: ITyX Contex Server
### END INIT INFO
# Do not modify anything beyond this point
```

Wie bei der Windows-Variante werden diese mithilfe eines 3rd-Party-Wrappers (Tanuki Wrapper) gestartet. Jedes Script muss an das System angepasst werden. Die absolut notwendigen Parameter der Start-Scripts sind `baseDir` und `pidDir`. Der User, der dieses Start-Script verwendet, muss Rechte auf beide Verzeichnisse haben.

### Komplette Liste

Parameter	Beschreibung
<code>baseDir</code>	Installations-Verzeichnis von Mediatrix-Contex (Absoluter Pfad).
<code>pidDir</code>	Verzeichnis, in dem die PID gespeichert werden soll.
<code>USER</code>	Name des Linux-Users, der dieses Script starten wird.
<code>APP_NAME</code>	Name des Daemons, der in der Prozessliste geführt wird.
<code>APP_LONG_NAME</code>	Beschreibung des Daemons.
<code>WRAPPER_CMD</code>	Pfad zur ausführbaren wrapper-Executable, wenn <code>baseDir</code> korrekt angegeben wurde, ist hier keine Änderung notwendig.
<code>WRAPPER_CONF</code>	Pfad zur ausführbaren wrapper-Konfiguration, z.B. <code>/conf/service/ityx-Contex-server.conf</code> .
<code>PRIORITY=</code>	Priorität des Daemons im Prozessmanager, entspricht der nice-Syntax.
<code>PIDDIR</code>	Entspricht <code>pidDir</code> .
<code>WAIT_AFTER_STARTUP</code>	Verzögert den Dienste-Start um X Sekunden.
<code>DETAIL_STATUS</code>	Kann <code>true</code> oder <code>false</code> sein. Wenn <code>true</code> , dann werden bei den <code>START</code> und <code>STOP</code> Kommandos mehr Informationen ausgegeben.
<code>RUN_AS_USER</code>	Sollte diese Variable gesetzt werden, so wird das Start-Script versuchen mittels <code>su</code> die Java-VM über einen bestimmten Linux-User zu starten.
<code># Do not modify anything beyond this point</code>	Über diese Zeile hinaus sollte das Script möglichst nicht mehr verändert werden!

### 3.7 Starten der Software

Die Dateien zum Starten der Software befinden sich unter `c:\mediatrix\bin\`. Dienste lassen sich auch in einer DOS-Box starten, eleganter ist aber die Dienste-Verwaltung des Betriebssystems zu verwenden.

---

*Ab Mediatrix v2.2 muss der Dienst „ITyX Licence-Service“-Dienst laufen, damit sich die Mediatrix-Dienste starten lassen.*

---

Alternativ zum einzelnen Start der Dienste als Applikation oder Service wurden Scripts vorbereitet, die die Dienste in der richtigen Reihenfolge und mit angemessenem Zeitabstand starten / stoppen

Script	Beschreibung
<b>Bin/00_all_context_services</b>	Startet die Dienste über DOS-Box (in der aktuellen Nutzer-Session) <ul style="list-style-type: none"> <li>- ActiveMQ</li> <li>- IDig Index Server</li> <li>- License Server</li> <li>- Context Server</li> <li>- Scheduler</li> <li>- ICat Assembler</li> <li>- ITag Trainer</li> </ul>

Wir stellen auch Startscripts für die Nutzung von Services / Daemons zur Verfügung. Diese sind:

Script
<b>Bin/service/start_all_context_services</b>
<b>Bin/service/start_all_services</b>
<b>Bin/service/stop_all_context_services</b>
<b>Bin/service/stop_all_services</b>

## 3.8 Clients starten

### 3.8.1 Context Admin starten



Abbildung 8 – Context Admin Client

Beim Start von Context Admin (`bin/03_contextadmin.*`) wird ein Login verlangt. In zukünftigen wird dieser Security Layer dazu dienen einzelne Änderungen in der Konfiguration nachvollziehbar zu machen. Der Standard-Login ist:

User:            admin

Passwort:       ityx

Benutzer-Konten können über diese Datei editiert werden:

`\conf\login-service\internal-users.xml`

### 3.8.2 Mediatrix-Client starten

Damit der Mediatrix-Client startfähig ist, muss der ITyX AppServer (der Applikationsserver) aktiv sein. Über das Start-Script `02_client.cmd` in `c:\mediatrix\bin` lässt sich der Client initialisieren. Es handelt sich hier um ein Java-Webstart-Programm, welches vom Applikationsserver heruntergeladen und gestartet wird. Der Client installiert sich auf diese Weise selbständig auf jeder Workstation, die einen Netzwerkzugriff zum Rechner besitzt, auf dem der Applikationsserver läuft. Um ihn über das Netzwerk zu starten, ist folgende Adresse im Browser aufzurufen:

<https://mediatrix.meinNetzwerk.local:8080/>

8080 ist der Port, auf den der Applikationsserver gebunden ist. Wenn Ihr Netzwerk über eine Firewall verfügt, muss dieser Port sowie Port 80 (http) offen sein.



Abbildung 9: Mediatrix Client Login

Nach erfolgreichem Starten der Software erscheint das Anmeldefenster. Die voreingestellten Zugangsdaten lauten

User: admin

Passwort: ityx

---

*Für die Applikationen und Clients wird ein X-Server (KDE, Gnome etc.) benötigt. Ein auf Linux gestarteter Mediatrix-Applikationsserver kann auch Windows-Clients bedienen (und umgekehrt).*

---

## 4 Allgemeine Systemkonfiguration

### 4.1 Übersicht der Verzeichnisstruktur

Verzeichnis	Beschreibung
/activemq/	Ein Dienst, der für die Funktion der License- und Scheduler-Dienste notwendig ist.
/bin/	Sammlung mit Start-Skripts für Windows und Linux. Programme wie Contex-Admin und Contex-Service-Center können hier gestartet werden. Auf Wunsch können auch die Dienste im Applikationsmodus gestartet werden, um z.B. Logausgaben in Echtzeit zu beobachten. Gemeinsame Variablen wie die Festlegung des Clusternamens müssen in der commonincludes.cmd konfiguriert werden.
/bin/icon/	Enthält *.ico-Dateien, um Desktop-Verknüpfungen mit ITyX-Symbolen auszustatten.
/bin/install/	Enthält Scripts, um ITyX-Dienste zu (de-)installieren und die Datenbanken, die in den Properties-Dateien eingetragen wurden zu installieren bzw. up-zu-daten.
/bin/logs/	Enthält Logdateien aller ITyX-Services, die als Applikation gestartet wurden. Format, Größe und Rollierung der Logdateien kann in den Dateien mediatrix.properties und ityxcommon.inc.conf angepasst werden  In Mediatrix aufgetretene INCIDENTS legen in diesem Verzeichnis HTML-Dateien ab
/bin/service/	Enthält 3rd-Party-Software, um die Installation von Diensten auf verschiedenen Plattformen (z.B. Windows 64 oder Linux) zu gewährleisten.
/conf/	Enthält globale Konfigurationen zu Log-Dateien, Datenbankverbindungsverwaltung und Cluster-Einstellungen. Die hier enthaltenen mediatrix.properties und contex.properties konfigurieren zum einen den Lizenz-Server und die sogenannte „Default“-Unit (das Basis-System).
/conf/custom/	Standard-Verzeichnis für eigene Konfigurationsdateien, z.B. für Business Rules und auch Mediatrix-Client-Einstellungen.
/conf/login-service/	Die hier befindlichen Konfigurationsdateien steuern den Login des Programms „Contex Admin“ – so wird etwa SSO über Kerberos möglich.  Hinweis: SSO über Contex und Mediatrix ist unterschiedlich zu konfigurieren

/conf/service/	Enthält Konfigurationen für Mediatrix- und Contex-Dienste (Windows-Dienste und Linux-Daemons gleichermaßen).
/data/	Enthält temporäre Daten der Contex-Dienste sowie Index-Dateien der Module „IDIG“ und „FUZZYMATCHER“
/es/	Enthält die Konfiguration zur ITyX Indizierung (IDig-Index, Fuzzymatcher)
/init-scripts/	Enthält Linux-Shell-Scripts, um Dienste unter Linux zu installieren.
/libs/	Das Libraries-Verzeichnis von Mediatrix, Contex und Ihrem Customizing (falls vorhanden)
/libs/classes/	Enthält eine Liste von Dateien in einer Text-Datei, die im Mediatrix-Client geladen sollen. Ändern Sie diese Datei nur dann, wenn Sie wissen, was Sie tun.
/libs/clientlibs/	Enthält Jar-Dateien, die den Mediatrix- und Contex-Services sowie dem Mediatrix-Client zur Verfügung gestellt werden (Customizing)
/libs/common/	Enthält Jar-Dateien, zu denen Mediatrix und Contex eine Abhängigkeit hat. Das Verzeichnis sollte nur dann Änderungen erfahren, wenn Sie wissen was Sie tun.
/libs/customlibs/	Enthält Jar-Dateien, die nur den Mediatrix- und Contex-Services zur Verfügung gestellt werden (Customizing)
/libs/db/	Enthält den Datenbank-Treiber im Jar-Format. Diesen Treiber müssen Sie als Kunde selbst herunterladen.
/libs/mediatrix/	Enthält die Programm-Dateien von allen ITyX-Diensten (Mediatrix und Contex).
/libs/nativelibs/	Enthält Plugins für die Nutzung von weiteren ITyX-Diensten, z.B. WebScout. Diese werden auch Mediatrix-Client zugänglich gemacht.
/licenses/	Enthält Lizenz-Informationen für alle Libraries, die im /libs/common/-Verzeichnis abgelegt wurden
/licenses-services/	Programm- und Konfigurationsdateien des ITyX License Services
/logs/	Enthält Logdateien aller ITyX-Services, die als Dienst gestartet wurden. Format, Größe und Rollierung der Logdateien kann in den Dateien mediatrix.properties und ityxcommon.inc.conf angepasst werden
/ocr/	Enthält Schnittstellen zu OCR-Software-Komponenten, um diese für den Contex-Prozess nutzbar zu machen. Standardmäßig liefern wir nur Tesseract3 aus, es sind aber auch Plugins für Abbyy Finereader und Omnipage verfügbar.

/tmp/	Enthält temporäre Dateien für Mediatrix-WebApp-Server, z.B. embedded Images von Textbausteinen. Löschen Sie diese Dateien nur dann, wenn Sie wissen was Sie tun.
/tomcat/	Tomcat-8-Verzeichnis; die Basis für die Ausführung von Mediatrix-WebApp-Server. Über die Tomcat-Konfiguration können Sie unter anderem den Port für Mediatrix-WebApp-Server konfigurieren.

## 4.2 Start-Scripts

Die Start-Scripts unterstützen Sie die Software möglichst einfach mit den gängigsten Konfigurationen zu starten. Die Scripts können angepasst werden, um die Software an Netzwerk und andere Anforderungen anzupassen.

### 4.2.1 /bin/

Script	Beschreibung
00_all_context_services	Startet die folgenden Dienste: ActiveMQ, Context-Indizierungs, License-Tracker, Context-Server, Scheduler, ICatAssembler, ITag Trainer Service
00_licensemanager	Startet die Applikation „License Manager“, um Lizenzen einzusehen und zu verwalten. Das Programm muss unter Windows bzw. einem X-Server gestartet werden.
00_license-service	Startet den License-Tracker als Applikation ohne die Context-Scheduler-Komponente.
00_managementserver	Startet den License-Tracker als Applikation mit der Context-Scheduler-Komponente.
01_webappserver	Startet den Mediatrix-Applikationsserver-Dienst als Applikation.
02_client	Startet einen Mediatrix-Client. Das Script beinhaltet den Aufruf von Java-Webstart sowie der URL zu Mediatrix (Localhost).
03_contextadmin	Startet den Context-Admin.
04_contextserver	Startet den Context-Server-Dienst als Applikation.
05_contextvagent	Startet die Virtual-Agent-Applikation, welche nötig ist, um auf diesem Rechner V-Agent-States auszuführen.



06_icat assembler	Startet den iCat-Assembler-Dienst als Applikation.
08_mediatrix_email_demon	Startet den Mediatrix-Email demon-Dienst als Applikation.
09_mediatrix_ptme	Startet den Mediatrix-PTME-Dienst als Applikation.
10_mediatrix_escalation_demon	Startet den Mediatrix-Escalation demon-Dienst als Applikation.
11_mediatrix_outbound_demon	Startet den Mediatrix-Outbound-Dienst als Applikation.
13_contextidigassembler	Startet den Context-IDig-Assembler-Dienst als Applikation.
14_report_generator	Startet den Mediatrix-Scheduled-Reports-Dienst als Applikation.
15_servicecenter	Startet das Context-ServiceCenter.
17_mediatrix_archive_scheduler	Startet den Mediatrix-Archive-Scheduler-Dienst als Applikation.
18_scheduler	Startet den Context-Scheduler-Dienst als Applikation.
19_idigindexserver	Startet den Context-Indizierungs-Dienst als Applikation.
20_iformassembler	Startet den Context-IForm-Dienst als Applikation.
20_itag_trainer_service_starter	Startet den Context-ITag-Trainer-Service-Dienst als Applikation.
21_dbmonitoring	Startet den Datenbank-Monitoring-Dienst als Applikation.
21_vocr_trainer_service_starter	Startet den VOCR-Trainer-Dienst als Applikation.
common_includes	Beinhaltet Parameter für alle Scripts in diesem Verzeichnis. <b>Diese Datei muss passend zu Ihrem System konfiguriert sein.</b>
start	Zentrales Steuerelement für alle Scripts in diesem Verzeichnis. Führen Sie das Start-Script ohne Parameter aus, um eine Auflistung der Optionen zu erhalten.
tools_pop3emlserver	Startet einen Minimalen POP3-Server, um EML-Dateien nach Mediatrix zu importieren.

#### 4.2.2 /bin/install/

Script	Beschreibung
--------	--------------

00_installService	Hilfsscript, um unter Windows ITyX-Dienst zu registrieren. Administrator-Rechte werden benötigt.
00_uninstallService	Hilfsscript, um unter Windows ITyX-Dienst zu entfernen. Administrator-Rechte werden benötigt.
01_contextDBinit	Start-Script zum Update und zur Erstellung der Context-Datenbank.
02_contextDBUpdate	Start-Script zum Update der Context-Datenbank.
04_mediatrixupdate	Start-Script zum Update und zur Erstellung der Mediatrix-Datenbank.
05_updateContextSubcats	Hilfsprogramm zur Zuordnung von Context-Mastern und Subcats zu Mediatrix-Projekten. DEPRECATED.
07_mediatrix_migration	Hilfsprogramm um Mediatrix-Datenbanken jeder Version auf die Datenbankstruktur von v2.2 anzupassen. Die Ausführung kann sehr lange dauern, jedoch kann das Script auch im laufenden Betrieb von Mediatrix durchgeführt werden, ohne dass Daten- oder Performanz-Verlust droht.
07_mediatrix_migration22	Datenbank-Struktur-Migration für Upgrades auf Mediatrix 2.2 oder höher, ausgehend von einer Version < 2.2. Wenn die Ausgangsversion höher oder gleich v2.2 ist, muss dieses Script nicht ausgeführt werden.
08_prepareWhiteKey	Vorgangsregistrierung für Upgrades auf Mediatrix 2.2 oder höher, ausgehend von einer Version < 2.2. Wenn die Ausgangsversion höher oder gleich v2.2 ist, muss dieses Script nicht ausgeführt werden. Alte Vorgänge werden innerhalb des License Managers registriert, da es vor v2.2 diese Funktion noch nicht gab.
09_mediatrix_migration_23	Datenbank-Struktur-Migration für Upgrades auf Mediatrix 2.3 oder höher, ausgehend von einer Version < 2.3. Wenn die Ausgangsversion höher oder gleich v2.3 ist, muss dieses Script nicht ausgeführt werden.
10_mediatrix_migration_24	Datenbank-Struktur-Migration für Upgrades auf Mediatrix 2.4 oder höher, ausgehend von einer Version < 2.4. Wenn die Ausgangsversion höher oder gleich v2.4 ist, muss dieses Script nicht ausgeführt werden.

11_mediatrix_migration_25	Datenbank-Struktur-Migration für Upgrades auf Mediatrix 2.5 oder höher, ausgehend von einer Version < 2.5. Wenn die Ausgangsversion höher oder gleich v2.5 ist, muss dieses Script nicht ausgeführt werden.
---------------------------	---

#### 4.2.3 /init-scripts/

Unter Linux werden keine Daemons registriert, sondern Shell-Scripts zu unterschiedlichen Runlevel hinzugefügt. ITyX stellt dafür Start-Scripts zur Verfügung, die entweder in die ServiceD oder /etc/init.d/-Umgebung eingebettet werden können. Jedes dieser Scripts muss ggf. an Ihr System angepasst werden.

Script	Beschreibung
ityx-Context-icatassembler	Linux-Shell-Script zum Starten des Dienstes Context ICat-Assembler
ityx-Context-idigassembler	Linux-Shell-Script zum Starten des Dienstes Context IDig-Assembler
ityx-Context-idigmaster	Linux-Shell-Script zum Starten des Dienstes Context IDig-Master
ityx-Context-server	Linux-Shell-Script zum Starten des Dienstes Context-Server
ityx-Context-workflow	Linux-Shell-Script zum Starten des Dienstes Context-Workflow
ityx-elasticsearch	Linux-Shell-Script zum Starten des Dienstes Elastic Search
ityx-licensetracker	Linux-Shell-Script zum Starten des License-Tracker-Dienstes
ityx-mediatrix-email	Linux-Shell-Script zum Starten des Dienstes Mediatrix-Emaildaemon
ityx-mediatrix-escalation	Linux-Shell-Script zum Starten des Dienstes Mediatrix-Escalation
ityx-mediatrix-outbound	Linux-Shell-Script zum Starten des Dienstes Mediatrix-Outbound
ityx-mediatrix-ptme	Linux-Shell-Script zum Starten des Dienstes Mediatrix-PTME
ityx-mediatrix-reportscheduler	Linux-Shell-Script zum Starten des Dienstes Mediatrix-Report-Scheduler
ityx-mediatrix-webapp	Linux-Shell-Script zum Starten des Dienstes Mediatrix-App-Server.

#### 4.2.4 Update-Scripts

Mediatrice und Context stehen Tools zur Verfügung, um die Datenbanken der Systeme zu installieren und auf dem neusten Stand zu halten. Die Tools sind unter `mediatrix-Context/bin/install/` zu finden. Sie heißen:

- `01_contextDBinit.cmd` / `01_contextDBinit.sh`
- `02_contextDBUpdate.cmd` / `02_contextDBUpdate.sh`
- `04_mediatrixUpdate.cmd` / `04_mediatrixUpdate.sh`

---

*contextDBinit ist "das alte ContextUpdate Script". Es sollte für Neuinstallationen und Updates innerhalb derselben Minor-Update-Version verwendet werden, z.B. von 2.5.4 auf 2.5.7*

*contextDBUpdate ist ein Migrations-Script für Context, welches ab 2.3 verfügbar ist. Wenn also größere Updates unternommen werden, z.B. ein Update von 2.4 auf 2.5, dann ist dieses Script zu verwenden. Die Eingabe bezüglich des Stands der Datenbank wäre in diesem Beispiel „2.4“.*

---

Voraussetzung für die erfolgreiche Durchführung eines Datenbankupdates ist eine korrekt eingetragene Datenbank in den Dateien `mediatrix.properties` und `context.properties`.

### 4.3 Kommunikation zwischen den Komponenten

#### 4.3.1 Externe Kommunikation zwischen den Komponenten

Die folgenden Kommunikationsprotokolle werden von Mediatrice und Context benutzt, die Ports können jeweils konfiguriert werden:

- IMAPS
- SMTP
- JDBC
- RMI
- http

#### 4.3.2 Interne Kommunikation zwischen den Komponenten

In der folgenden Abbildung ist dargestellt wie die einzelnen Komponenten des Systems untereinander kommunizieren.

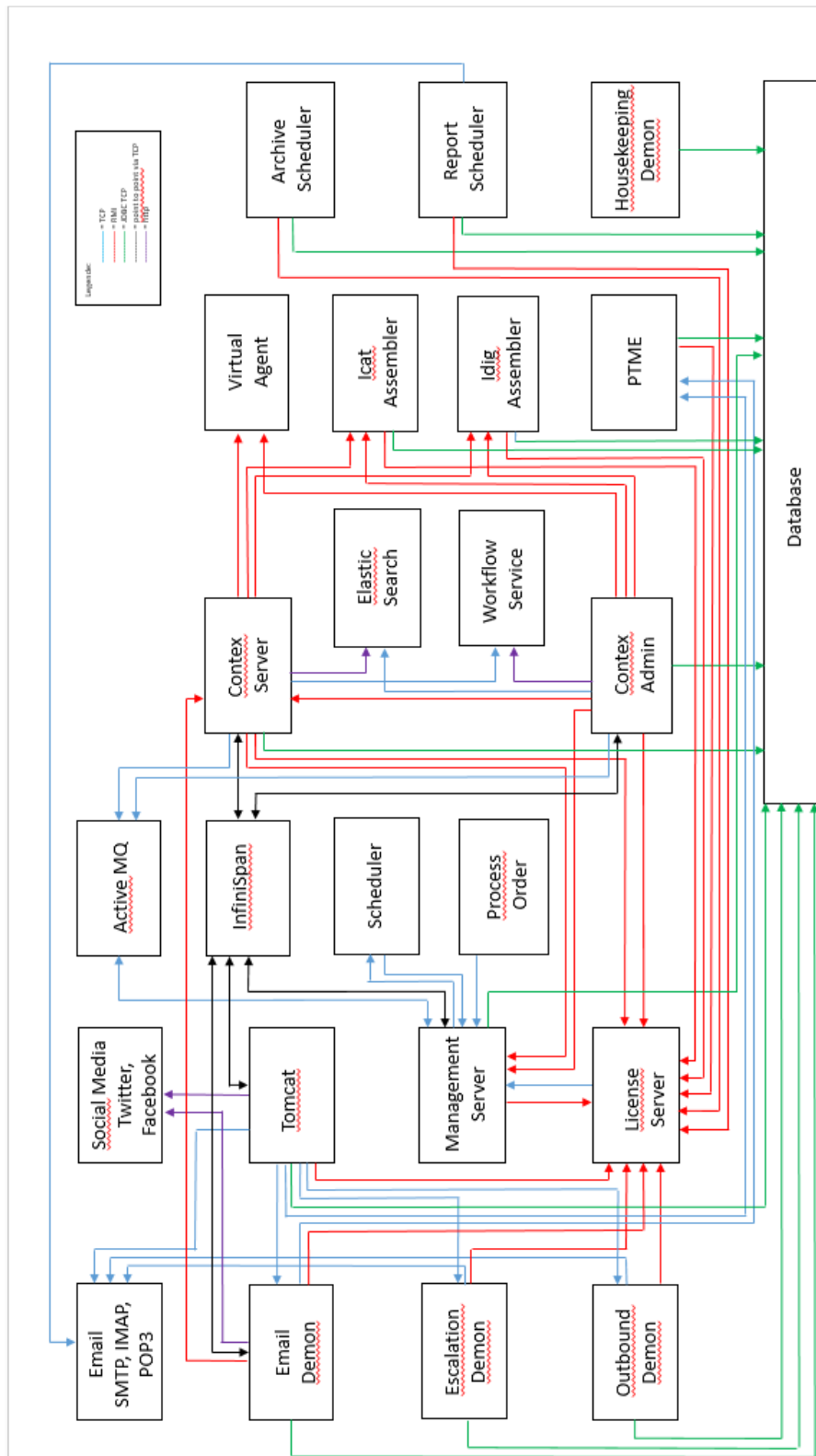


Abbildung 10 – Interne Kommunikation

#### 4.3.3 Liste der Protokolle und Ports

Dies ist eine Zusammenstellung aller Ports, die in Mediatrix verwendet werden können.

Komponente		Protokoll	Port
<b>Tomcat (webapp)</b>			
- PTME	out	TCP	port 6413 konf
- Contextserver	out	RMI	registry port 1099 konf, object port dyn
- Licenser	out	RMI	registry port 1099 konf, object port dyn
- Emaildemon	out	TCP	port 17121 konf
- Escalationdemon	out	TCP	port 17122 konf
- Outbounddemon	out	TCP	port 17123 konf
- InfiniSpan(JGroups)	out	PtP dyn über TCP	UDP-port konf, MultiCast Adresse konf
- DB	out	JDBC TCP	port konf
- Email SMTP,IMAP,POP3 (+s)	out	IMAP	port konf
- Social Media (Twitter, Facebook)	out	HTTP(S)	port 80 bzw. 443
- Status Port (Cache reset, Logausgaben)	in	TCP	port 17125 konf
- JMX	in	RMI	port 9100
- Servicecontrolserver	in	RMI	9011 konf
- InfiniSpan(JGroups)	in	PtP dyn über TCP	UDP-port konf, MultiCast Adresse konf
<b>Emaildemon</b>			
- DB	out	JDBC TCP	port konf
- Email SMTP,IMAP,POP3 (+s)	out	IMAP	port konf
- Social Media (Twitter, Facebook)	out	HTTP(S)	port 80 bzw. 443
- PTME	out	TCP	port 6413 konf
- Contextserver	out	RMI	registry port 1099 konf, object port dyn
- Licenser	out	RMI	registry port 1099 konf, object port dyn
- InfiniSpan(JGroups)	out	PtP dyn über TCP	UDP-Port konf, MultiCast Adresse konf

- JMX	in	RMI	port 9160 konf
<b>Escalationdemon</b>			
- DB	out	JDBC TCP	port konf
- Email SMTP (+s)	out	SMTP(S)	port konf
- Licensesserver	out	RMI	registry port 1099 konf, object port dyn
- JMX	in	RMI	port 9170 konf
<b>Outbounddemon</b>			
- DB	in	JDBC TCP	port konf
- Email SMTP (+s)	in	SMTP(S)	port konf
- Licensesserver	in	RMI	registry port 1099 konf, object port dyn
- JMX	in	RMI	port 9180 konf
<b>PTME</b>			
- DB	out	JDBC TCP	port konf
- JMX	in	RMI	port 9190 konf
- Licensesserver	out	RMI	registry port 1099 konf, object port dyn
<b>Archivscheduler</b>			
- DB	out	JDBC TCP	port konf
- JMX	in	RMI	N/A
- Licensesserver	out	RMI	registry port 1099 konf, object port dyn
<b>Reportscheduler</b>			
- DB	out	JDBC TCP	port konf
- Email SMTP (+s)	out	SMTP(S)	port konf
- JMX	in	RMI	port 9210
- Licensesserver	out	RMI	registry port 1099, object port dyn

<b>Housekeepingdemon</b>			
- DB	out	JDBC TCP	port konf
<b>MX-Client</b>			
- Webapp	out	HTTP(S)	port 8180 konf
<b>Managementserver</b>			
- ProcessOrder			
- Scheduler			
- License Server			
- Library			
- DB	out	JDBC TCP	port konf
- Active MQ	out	TCP	port 61616 konf
- InfiniSpan(JGroups)	out	PtP dyn über TCP	UDP-Port konf, MultiCast Adresse konf
- Scheduler	out	RMI	1099 konf , objectport dyn
- FTP(S)		FTP	konf
- SFTP		SSH	konf
- IMAP(S)		IMAP(s)	konf
- POP3(s)		POP3(S)	konf
- SCP		SSH	konf
- WEBDAV(http)		HTTP	konf
- License Server	out	RMI	registry port 1099, object port dyn
- JMX	in	RMI	port 9230 konf
- Active MQ	in	TCP	port 61616 konf
- InfiniSpan(JGroups)	in	PtP dyn über TCP	UDP-port konf, MultiCast Adresse konf



<b>Contexserver</b>			
- ICat	in	RMI	registry port 1099 konf, object port dyn
- VCat	in	RMI	registry port 1099 konf, object port dyn
- MCat	in	RMI	registry port 1099 konf, object port dyn
- Extag	in	RMI	registry port 1099 konf, object port dyn
- FuzzyMatcher	in	RMI	registry port 1099 konf, object port dyn
- Workflow	in	RMI	registry port 1099 konf, object port dyn
- OCR	in	RMI	registry port 1099 konf, object port dyn
- ITag	in	RMI	registry port 1099 konf, object port dyn
- IForm	in	RMI	registry port 1099 konf, object port dyn
- ExportService	in	RMI	registry port 1099 konf, object port dyn
- IDig	in	RMI	registry port 1099 konf, object port dyn
- Docpool	in	RMI	registry port 1099 konf, object port dyn
- DB	out	JDBC TCP	port konf
- Licensesserver	out	RMI	registry port 1099 konf, object port dyn
- Contexserver	out	RMI	registry port 1099 konf, object port dyn
- Managementserver	out	RMI	registry port 1099 konf, object port dyn
- InfiniSpan(JGroups)	out	PtP dyn über TCP	UDP-port konf, MultiCast Adresse konf
- Active MQ	out	TCP	port 61616 konf
- Workflowservice	out		konf
- FTP(S)		FTP	konf
- SFTP		SSH	konf
- IMAP(S)		IMAP(s)	konf
- POP3(s)		POP3(S)	konf
- SCP		SSH	konf

- SMTP		SMTP	konf
- WEBDAV(http)		HTTP	konf
- VAgent	out	RDP / RMI	3389 / 1099 konf
- Elasticsearch	out	HTTP	konf zwischen den Nodes
- Virtual Agent	out	RMI	RDP konf
- ICatassembler	out	RMI	port 1099 konf
- IDigassembler	out	RMI	port 1099 konf
- JMX	in	RMI	port 9140 konf
- InfiniSpan(JGroups)	in	PtP dyn über TCP	UDP-port konf, MultiCast Adresse konf
<b>Contex-Admin</b>			
- DB	out	JDBC TCP	port konf
- Licenseserver	out	RMI	registry port 1099 konf, object port dyn
- Contexserver	out	RMI	registry port 1099 konf, object port dyn
- Managementserver	out	RMI	registry port 1099 konf, object port dyn
- InfiniSpan(JGroups)	out	PtP dyn über TCP	UDP-port konf, MultiCast Adresse konf
- Active MQ	out	TCP	port 61616 konf
- Workflowservice	out	RMI	1099 konf, object port dyn
- FTP(S)		FTP	konf
- SFTP		SSH	konf
- IMAP(S)		IMAP(s)	konf
- POP3(s)		POP3(S)	konf
- SCP		SSH	konf
- SMTP		SMTP	konf
- WEBDAV(http)		HTTP	konf
- HTTP	out	HTTP	80 konf
- VAgent	out	RDP / RMI	3389 / 1099 konf
- Elasticsearch	out	HTTP	konf zwischen den Nodes
- Virtual Agent	out	RMI	RDP konf
- Icatassembler	out	RMI	port 1099 konf
- IDigassembler	out	RMI	port 1099 konf

- InfiniSpan(JGroups)	in	PtP dyn über TCP	UDP-port konf, MultiCast Adresse konf
<b>ElasticSearch</b>			
- Elasticsearch	out	HTTP	konf zwischen den Nodes
- Elasticsearch	in	HTTP	konf zwischen den Nodes
<b>ICatassembler</b>			
- DB	out	JDBC TCP	port konf
- Licensesserver	out	RMI	registry port 1099 konf, object port dyn
- Contexserver	in	RMI	registry port 1099 konf, object port dyn
- ContexAdmin	in	RMI	registry port 1099 konf, object port dyn
- JMX	in	RMI	port 9110
<b>IDigassembler</b>			
- DB	out	JDBC TCP	port konf
- Licensesserver	out	RMI	registry port 1099 konf, object port dyn
- Contexserver	in	RMI	registry port 1099 konf, object port dyn
- ContexAdmin	in	RMI	registry port 1099 konf, object port dyn
- JMX	in	RMI	9120 konf
<b>Virtual Agent</b>			
Vagent Demon	in	RMI	registry port1099 konf , objectport dyn
<b>AbbyyConnector</b>			

- AbbyyLicenseService	In	TCP	port 3000 konf
- AbbyyConnector	out	TCP	port3000 konf

#### 4.3.4 Zuordnung JVM und Komponenten

Jedes Startscript startet eine JVM in der dann die aufgeführten Komponenten laufen:

Startscript	Komponente
00_licensemanager.cmd /sh	License Manager GUI
00_license-service.cmd /sh	ProcessOrder LicenseServer LibraryManger
00_managementserver.cmd /sh	ManagementServer ProcessOrder Scheduler LicenseServer LibraryManager
webappserver.cmd /sh	webapp (MediatrixServer)
02_client.cmd / sh	MediatrixClient
03_contexadmin.cmd	ContexAdmin GUI
04_contexserver.cmd / sh	ICat (Categorizer) VCat(Visual Categorizer) MCat(Manual Categorizer) Extag (Extraction) FuzzyMatcher (Fuzzy Search) WorkflowEngine OCR Server

	ITag (Extraction) IForm ( Visual Categorizer) ExportService IDIG (Crawler) DocpoolService
05_contextvagent.cmd / sh	Contex VAgent
06_icatassembler.cmd / sh	IcatAssembler (ICat Trainer)
08_mediatrix_email_demon.cmd	Mediatrix EmailDemon
09_mediatrix_ptme.cmd / sh	PTMEServer
10_mediatrix_escalation_demon.cmd / sh	EscalationDemon
11_mediatrix_outbound_demon.cmd / sh	OutboundDemon (Kampangen)
13_contextidigassembler.cmd / sh	IDig Crawler
14_report_generator.cmd / sh	Reporting
15_servicecenter.cmd / sh	ServiceCenter GUI (Start/Stop von Workflows, DocPool, Monitoring der Workflows und der einzelnen Prozessschritte)
16_mediatrix_import_user.cmd / sh	UserImport
17_mediatrix_archive_scheduler.cmd / sh	ArchiveScheduler
18_scheduler.cmd / sh	Scheduler
19_idigindexserver.cmd / sh	ElasticSearch (FuzzyMatcher)

## 4.4 Konfiguration mediatrix.properties

Dieses Kapitel liefert eine Übersicht aller Konfigurationsoptionen, die in der globalen Konfigurationsdatei (`mediatrix.properties`) aller ITyX-Mediatrix-Dienste gesetzt werden können. Die Datei befindet sich im Verzeichnis `mediatrix-Contex/conf/`. Diese Liste erhebt den Anspruch auf Vollständigkeit für alle im Basis-System von Mediatrix implementierten Optionen. D.h. wenn sich ein Eintrag nicht in dieser Liste befindet, so kommt diese entweder aus einem Kundenmodul (Customizing) oder ist „DEPRECATED“. Die Markierung „DEPRECATED“ bedeutet, dass diese Option nicht mehr unterstützt wird (wenn nicht jetzt schon, dann in einer der nächsten Versionen).

### 4.4.1 Datenbank-Einstellungen

Die folgenden Einstellungen beziehen sich auf den Datenbankzugriff durch die Dienste Mediatrix WebApp, Emailedemon, Outbound, Escalation und PTME. Speziell die Zugangsdaten zur Datenbank sind sorgfältig zu prüfen, da ohne eine intakte Datenbankverbindung die Software nicht starten kann.

#### *betweenruns*

Beispiel: `betweenruns=<Zahl>` (default 600)

Zeitintervall in Sekunden, das zwischen den Überprüfungen auf unbenutzte Datenbankverbindungen liegt. Der Wert sollte die Einstellung im SQL-Server (suche alle X Minuten nach unbenutzten SQL-Verbindungen und schließe diese) nicht überschreiten, da es sonst zu SQL-Fehlern im Client kommt, die zwar nicht gefährlich, aber störend sind. Diese Einstellung ist in Verbindung mit `connectionpool=1` und `poollifetime` nutzbar.

#### *connectionpool*

Beispiel: `connectionpool=1` (default 0)

Aktiviert die Funktion des Connectionpools. Diese Option hält die Anzahl der Datenbankverbindungen auf einem geringen Niveau. Es ist empfehlenswert, diesen Schalter in Verbindung mit `poolmaxactive` zu verwenden (Hinweis: in älteren ausgelieferten Konfigurationsdateien steht noch `maxactive`, das ist leider falsch). `servletsynchronized` sollte auf 0 gestellt werden, wenn die Option `connectionpool=1` gesetzt wird - und umgekehrt. Die beiden Einstellungen sollten immer komplementär zueinander sein.

#### *connectionpooldebug*

Beispiel: `connectionpooldebug=1` (default 0)

Aktiviert erweiterte Debug-Logausgaben für die Verwendung des Connectionpools.

#### *dbappendix*

`dbappendix=<db-type>` (kein default-Wert)

Mögliche Optionen sind: z.B. `mysql`, `mssql`, `oracle`, `mssql2005`, `mssql2005integrated`, etc).

Beispiel: dbappendix=mysql

Definiert, welches Datenbank-Management-System verwendet wird.

Hier eine Übersicht für Administratoren, welche Werte für „dbappendix“ vergeben werden können:

dbappendix	Beschreibung
mssql	Verwendbar ab MSSQL (in den meisten Fällen deckt dies alle Versionen von MSSQL ab (bis 2008R2))
mssql2005	Verwendbar ab MSSQL 2005
mssql2005special	Verwendbar ab MSSQL 2005  Die „alte“ Version von mssql2005, welche kein UNION ALL unterstützt (Performance-Verlust) und ohne nText-Tabellen ausführbar ist. Diesen dbappendix nur dann verwenden, wenn „mssql2005“ nicht funktionieren sollte.
mssql2005integrated	Siehe „mssql2005“, unterstützt zusätzlich die MSSQL Integrated Security
mssql2008	Verwendbar ab MSSQL 2008
mssql2008integrated	Siehe mssql2008, unterstützt zusätzlich die MSSQL Integrated Security
mssql20082integrated	Siehe „mssql2008integrated“, verzichtet jedoch darauf „selectMethod=cursor“ in die Connection-URL mit einzubeziehen
mssql2	Siehe „mssql“, nur mit dem Unterschied, dass kein DBMS-Dialect vorgegeben wird. Es wird der default-Wert des JDBC-Treibers verwendet
mssql20052	Siehe mssql2005, nur mit dem Unterschied, dass kein DBMS-Dialect vorgegeben wird. Es wird der default-Wert des JDBC-Treibers verwendet. mssql2005smcursor MSSQL 2005 Treiber, mit MSSQL-2005-Dialect und „selectMethod=cursor“
mscluster	MSSQL Cluster Databases Support
oracle	Verwendbar ab Oracle 8g (dbname muss auf eine SID zeigen)
oracle9	Verwendbar ab Oracle 9g (dbname muss auf eine SID zeigen)
oracle10	Verwendbar ab Oracle 10g (dbname muss auf eine SID zeigen)
oracle11	Verwendbar ab Oracle 11g (dbname muss auf eine SID zeigen)

dbappendix	Beschreibung
oracle11service	Verwendbar ab Oracle 11g (dbname muss auf einen DB-Service zeigen)
oracleRAC2	<p>Oracle Real Application Cluster Support (2 Server)</p> <p>Details zum OJDBC Connection String:</p> <pre> BASEURL "@(DESCRIPTION=(LOAD_BALANCE=OFF)(FAILOVER=ON) (ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(Host=@SERVER1@)(Port= @PORT1@ )) (ADDRESS=(PROTOCOL=TCP)(Host=@SERVER2@)(Port=@PORT2@))) (CONNECT_DATA=(SERVICE_NAME=" DATABASE ")))" </pre>
oracleRAC3	<p>Oracle Real Application Cluster Support (3 Server)</p> <p>Details zum OJDBC Connection String:</p> <pre> BASEURL "@(DESCRIPTION=(LOAD_BALANCE=OFF)(FAILOVER=ON) (ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(Host=@SERVER1@)(Port= @PORT1@ )) (ADDRESS=(PROTOCOL=TCP)(Host=@SERVER2@)(Port=@PORT2@)) (ADDRESS=(PROTOCOL=TCP)(Host=@SERVER3@)(Port=@PORT3@))) (CONNECT_DATA=(SERVICE_NAME=" DATABASE ")))" </pre>
mysql	Verwendbar ab MySQL v5.0
mysql5	Siehe „mysql“, es wird jedoch ein spezifischer MySQL-5-Dialect verwendet
mysql_unicode	<p>Verwendbar ab MySQL v5.0</p> <p>MySQL Character Encoding UTF-8 Support</p>



*dbhost*

dbhost=<IP-Adresse / DNS> (default 127.0.0.1)

Beispiel: dbhost=192.168.123.15

Adresse der Datenbank. Gültige Werte schließen einen DNS-Namen oder eine IP-Adresse ein. Es ist jedoch empfehlenswert eine feste IP-Adresse einzutragen, da je nach Operation der Software und Server der DNS-Server im Netzwerk mit vielen Requests belastet wird. Alternativ sollte der DNS in den lokalen hosts-Files eingetragen sein.

*dbname*

dbname=<String>

Beispiel: dbname=mediatrix

Name der Datenbank im DBMS. Bei Oracle Datenbanken ist hier die SID bzw. Service-Name des anzusprechenden Systems zu hinterlegen.

*dbuser*

dbuser=<String>

Beispiel: dbuser=user

Login des Datenbank-Users, der Zugriff auf die Datenbank (dbname) hat. Der User-Account muss Rechte für die Befehle SELECT, UPDATE, INSERT und DELETE haben.

*dbpassword*

dbpassword=<String>

Beispiel: dbpassword=password

Passwort des Datenbank-Users, der Zugriff auf die Datenbank (dbname) hat. Beginnt ein Passwort mit #+, so wird eine verschlüsselte Form des Kennwortes erwartet. Die Kennwort-Verschlüsselung kann im Hilfe-Menü des Mediatrix Clients aufgerufen werden.

*dbport*

dbport=<Zahl>

Beispiel: dbport=3306

TCP-Port, auf dem die Datenbank zu erreichen ist. Standardwerte sind 3306 (MySQL), 1433 (MSSQL) und 1521 (Oracle).

Hinweis: Wenn Instanzen von MSSQL verwendet werden, z.B. mssql\instanz01, muss geprüft werden, ob diese Instanz nicht vielleicht einen eigenen Port anstatt 1433 hat. Oft haben Instanzen einen dynamischen Port, der zufällig im Bereich 50000-60000 gewählt wird und beim Restart des Datenbank-Dienstes wieder anders sein kann. Es wird ausdrücklich empfohlen einen festen Port einzustellen (MSSQL Configuration Tool → DBInstanz → TCP/IP → Port).

#### *oracle.usesequence*

oracle.usesequence=1 (default 0)

Oracle Datenbanken beinhalten einen internen Sequencer. Falls es erwünscht ist diesen, anstatt des Mediatrix-internen Sequencers, zu verwenden, so muss dieser Schalter auf 1 gesetzt werden. Bislang gibt es keine Berichte, welcher der beiden Sequencers besser ist. Die Option sollte nur verwendet werden, falls Probleme in diesem Bereich auftreten.

#### *poolmaxactive*

poolmaxactive=<Zahl> (default 150)

Beispiel: poolmaxactive=80

Anzahl der Verbindungen zu einem Connectionpool. Diesen Schalter nur in Verbindung mit [connectionpool=1](#) verwenden.

#### *poollifetime*

poollifetime=<Zahl> (default 300)

Beispiel: poollifetime=450

Maximale Lebensdauer für eine unbenutzte Connection. Ist die Lebensdauer überschritten, wird die Connection bei der nächsten Überprüfung geschlossen.

#### *servletsynchronized*

servletsynchronized=0 (default 1)

Mit dieser Einstellung lässt sich das parallele Laden von Textbausteinvorschlägen aktivieren. Der Default-Wert ist 1 und bildet das alte Verhalten (v1.3) ab. [servletsynchronized](#) sollte auf 0 gestellt werden, wenn die Option [connectionpool=1](#) gesetzt wird - und umgekehrt. Die beiden Einstellungen sollten immer komplementär zueinander sein.

### 4.4.2 Allgemeines

#### *alternativecolors*

alternativecolors=1 (default 0)

Aktiviert ein alternatives Color-Set für den Mediatix-Client. Anstatt der Vordergrundfarbe wird der Hintergrund zur Markierung der Eskalationsstufe und vieles mehr umgefärbt.

#### *alternate.htmltype*

alternate.htmltype=false (default true)

Dieser Schalter wird z.B. in der Klasse AskMeEditorPane überprüft. Ist eine HTML-Mail nicht ordentlich darstellbar, weil Tags fehlen, so wird die Mail als Text dargestellt. Setzt man den Schalter auf false, so wird versucht, die HTML-Elemente darzustellen (Outlook-ähnliches Verhalten).

#### *altupdate*

altupdate=ping (default #auskommentiert)

Das Standardverhalten sendet alle 5 Sekunden ein Ping-Command an den Datenbank-Server als keepalive-Signal. Ab einer gewissen Menge von Usern ist die Last jedoch zu hoch. altupdate=ping sammelt die Ping-Anfragen und sendet das Update in einem String an den Datenbankserver.

#### *applog.emailinformation*

applog.emailinformation=1 (default 0)

Gibt die Email-Header und den Body komplett aus, standardmäßig geschieht das jetzt nicht mehr.

#### *attachment\_maxsize*

attachment\_maxsize=<Zahl><Einheit> (default 8m)

Beim Einfügen von Attachments in eine Email (Initial-Email oder Mediatix-Antwort), wird die Größe aller eingefügten Attachments addiert und mit dem „attachment\_maxsize“-Wert verglichen. Ist die Größe überschritten, wird der Benutzer über einen Dialog aufgefordert einzelne Attachments zu entfernen, bis der Wert unterschritten ist. Als Einheiten für diesen Schalter gelten:

m,M = Megabyte

k,K = Kilobyte

<leer> = Byte

#### *baseremotelogger*

baseremotelogger=1 (default 0)

Wenn baseremotelogger auf true steht (1), werden die Debugausgaben des Connectionpools auf dem Remoteport ausgegeben (telnet localhost 17121-17125).

*callDialog\_loadAllSubprojects*

callDialog\_loadAllSubprojects=1 (default 0)

Steht der Schalter auf 1, so werden beim Erstellen eines Telefonkontaktes die Zuständigkeiten vernachlässigt und alle Teilprojekte geladen, in denen der Kontakt dann auch erstellt werden darf.

*caseinsensitive*

caseinsensitive=1 (default 0)

Ermöglicht eine case-insensitive Volltext-Suche mit Oracle-Datenbanken. Ab einer gewissen Menge sollte in der Theorie die Performance aber in den Keller gehen, weil sie nicht über den Index sucht. Die Suche klappt zur Zeit für Fragen, Antworten und Vorgänge.

*case.transfer.keepfrom*

case.transfer.keepfrom=1 (default)

Ein Vorgang, der einem anderen Kunden zugeordnet wird, behält mit case.transfer.keepfrom=1 die originale From: ...- und To: ...-Mail-Adresse.

*checksend*

checksend=1 (default 0)

trägt eine Bestätigung ins Log ein, wenn eine Mail erfolgreich über den SMTP versendet wurde. Diese Einstellung sollte nur zu Debug-Zwecken verwendet werden.

*checklistvalidation*

checklistvalidation=hard (default soft)

„hard“ bedeutet, wenn eine Frage durch Absenden oder Erledigen abgeschlossen wird, passiert folgendes:

Es wird überprüft, ob die Checkliste komplett abgearbeitet wurde oder nicht

Wird ein Pflichtpunkt nicht abgehakt, ist eine Aktion (Erledigen oder Absenden) nicht möglich und es erscheint eine Warnung

Werden alle Pflichtpunkte abgehakt, aber nicht alle optionalen Pflichtpunkte, dann erscheint eine Meldung, dass die Checkliste nicht komplett abgearbeitet wurde und wird der Agent wird gefragt, ob man die Aktion fortsetzen möchte

mit „ja“ wird die Aktion fortgesetzt

mit „nein“ bleibt die Frage offen zur weiteren Bearbeitung

„soft“ bedeutet, dass, obwohl Pflichtpunkte nicht selektiert sind, die Aktion (erledigen oder absenden) fortgesetzt werden kann. Es erscheint eine Meldung mit der Frage, ob man weitermachen möchte

#### *client.change.language*

client.change.language=false (default true)

Wird dieser Schalter auf „false“ gestellt, so ist es Mitarbeitern nicht mehr möglich die Client-Sprache zu ändern. Es wird die Sprache forciert, die über den Schalter client.default.language konfiguriert wurde.

#### *client.default.language*

client.default.language=en\_GB (default de\_DE)

Diese Option legt die Standardsprache aller Mediatrix-Clients fest. Der Default-Wert ist „deutsch“. Zu verwenden sind Locale-Variablen, wie sie Java verwendet. Abhängig davon, in welcher Sprache Java JRE / SDK auf dem Mediatrix-Server installiert ist, können die Optionen dieses Schalters abweichen<sup>7</sup>.

#### *client.maxmb*

Beispiel: client.maxmb=1024 Diese Option legt den Heap-Space des Clients fest. Der Default-Wert liegt bei 720MB pro Client.

#### *client.propertypath*

client.propertypath=<String>

client.propertypath=../conf/custom/

Definiert den Pfad relativ zum Verzeichnis, von dem die Mediatrix-Dienste ausgeführt werden<sup>8</sup>. Es können auch absolute Pfade eingetragen werden<sup>9</sup>. In diesem Verzeichnis ist eine Datei namens mediatrix.rc zu hinterlegen. In die Datei können Java-Properties und Parameter eingetragen werden, die an alle Mediatrix-Client als zusätzliche Parameter weitergegeben werden.

#### *client.stage*

Beispiel: client.stage=Produktion

<sup>7</sup> Für eine Übersicht der „Java Locales“ siehe <http://www.oracle.com/technetwork/java/javase/locales-137662.html>

<sup>8</sup> Mediatrix Dienste werden über den Tanuki Java Wrapper ausgeführt. Dieser befindet sich unter mediatrix/bin/service/win64/ (der letzte Ordner ist abhängig vom Betriebssystem)

<sup>9</sup> Pfade unter Windows müssen mit „/“, nicht mit „\“, eingetragen werden

Mit Hilfe dieser Option lässt sich ein zusätzliches Suffix in den Fenster-Titel des Mediatrix-Clients setzen. Auf diese Weise kann z.B. einfach zwischen einer TEST- und PRODUKTIV-Umgebung unterschieden werden.

#### *client.vmargs*

Beispiel: `client.vmargs=-ea -Xincgc`

Mithilfe dieses Schalters können Mediatrix-Clients zusätzliche Java-Properties übergeben werden. Sie werden über die dynamisch erzeugten JNLP-Scripts publiziert, die immer dann generiert werden, wenn z.B. `http://localhost:8080/mediatrix/jnlp` aufgerufen wird, also bei jedem Client-Start. Innerhalb der JNLP werden diese zusätzlichen VM-Argumente in der Zeile „`<j2se version= ...>`“ verewigt, z.B. `<j2se version="1.4+"java-vm-args="-ea -Xincgc"/>`.

#### VM Args Kurzbeschreibung

<code>-d32</code>	use a 32-bit data model if available (unix platforms only)
<code>-client</code>	to select the client VM
<code>-server</code>	to select the server VM
<code>-verbose</code>	enable verbose output
<code>-version</code>	print product version and exit
<code>-showversion</code>	print product version and continue
<code>-help</code>	print this help message
<code>-X</code>	print help on non-standard options
<code>-ea</code>	enable assertions
<code>-enableassertions</code>	enable assertions
<code>-da</code>	disable assertions
<code>-disableassertions</code>	disable assertions
<code>-esa</code>	enable system assertions
<code>-enablesystemassertions</code>	enable system assertions

-dsa	disable system assertions
-disablesystemassertions	disable system assertions
-Xmixed	mixed mode execution (default)
-Xint	interpreted mode execution only
-Xnoclassgc	disable class garbage collection
-Xincgc	enable incremental garbage collection
-Xbatch	disable background compilation
-Xprof	output cpu profiling data
-Xdebug	enable remote debugging
-Xfuture	enable strictest checks, anticipating future default
-Xrs	reduce use of OS signals by Java/VM (see documentation)
-XX:+ForceTimeHighResolution	use high resolution timer
-XX:-ForceTimeHighResolution	use low resolution (default)

Folgende Argumente können Parameter mitgegeben werden, z.B. -XX:PermSize=128M.

#### VM Args + Parameter Kurzbeschreibung

-ea	enable assertions for classes
-enableassertions	enable assertions for classes
-da	disable assertions for classes
-disableassertions	disable assertions for classes
-verbose	enable verbose output
-Xms	set initial Java heap size
-Xmx	set maximum Java heap size
-Xss	set java thread stack size

-XX:NewRatio	set Ratio of new/old gen sizes
-XX:NewSize	set initial size of new generation
-XX:MaxNewSize	set max size of new generation
-XX:PermSize	set initial size of permanent gen
-XX:MaxPermSize	set max size of permanent gen
-XX:MaxHeapFreeRatio	heap free percentage (default 70)
-XX:MinHeapFreeRatio	heap free percentage (default 40)
-XX:UseSerialGC	use serial garbage collection
-XX:ThreadStackSize	thread stack size (in KB)
-XX:MaxInlineSize	set max num of bytecodes to inline
-XX:ReservedCodeCacheSize	Reserved code cache size (bytes)
-XX:MaxDirectMemorySize	A user-settable upper limit on the maximum amount of allocatable direct buffer memory

#### *combobox.report.type*

combobox.report.type=no (default yes)

Der Inhalt der Dropdown-Box „Report-Typ“ des Report-Generators (erreichbar über das Menü Statistiken) wird alphabetisch sortiert, wenn der Schalter auf 'yes' steht. Steht er auf 'no', so ist das ursprüngliche Verhalten aktiviert (unsortiert).

#### *connectioncheck*

connectioncheck=1 (default 0))

ist ein debug-Schalter, welcher die SQL-Statements abhängig vom Mitarbeiter in ein Log schreibt „Benutzer admin sql: select \* from mitarbeiter;“. Der Schalter sollte nicht dauerhaft eingeschaltet sein, weil die Logs dadurch EXTREM viele Einträge erhalten (=Last auf dem Server).

#### *contenttype.check*

contenttype.check=0 (default 1)

E-Mails, deren Header Attachments oder Inhalten einen falschen Content-Type zuweisen, werden über



diesen Schalter überprüft. Werden z.B. PDF-Dateien als HTML-Dateien deklariert, wird der Inhalt des PDFs von Mediatrix als HTML-Datei interpretiert. Ab Version 2.1.38 bzw. 2.0.107 wird diese Zuordnung automatisch geprüft, sollte die Zuordnung nicht mit dem tatsächlichen MIME-Type zusammenpassen, wird „application/octetstream“ als Content-Type gesetzt. Mit contenttype.check=0 wird das alte Verhalten wiederhergestellt.

#### *context.textobject.suggest*

Context.textobject.suggest=1 (default 0)

In der Regel werden Textbausteinvorschläge für Antworten innerhalb des Mediatrix-Clients durch die PTME erstellt. Für jede geöffnete Mail wird ein Thread erstellt, der anhand der gesammelten KI-Daten in der Datenbank die Vorschläge berechnet. Wenn dies nicht performant genug ist, kann die Context-Suche diese Funktion ersetzen. Voraussetzung, um das Verhalten erfolgreich umzustellen (=1), ist ein eingerichteter Information-Retrieval-Prozess, der die Mediatrix-Datenbank in regelmäßigen Abständen indiziert.

#### *context.icatport*

Beispiel: Context.icatport=1234 (default 1099)

Dieser Schalter zeigt auf den Port des Context Categorizer Services. Dieser ist in der Regel identisch mit dem Port des Context Services, da der „Categorizer“ ein Modul (icat) dieses Dienstes ist.

#### *context.propertypath*

Context.propertypath=/home/localhorst (default user.home)

Falls die context.properties nicht im Home-Verzeichnis liegt, weil der z.B. Kunde auf eine Trennung der Systeme besteht, kann man mit diesem Schalter den Pfad entsprechend anpassen.

#### *contextdb.directaccess*

Contextdb.directaccess=1 (default 0)

Wird dieser Schalter in der Datei mediatrix.properties gesetzt, ist es jedem Mediatrix-Client erlaubt eine direkte Verbindung mit der Context-Datenbank herzustellen. Auf diese Weise wird die ursprüngliche Technik, jedes Signal über den VJDBC-Treiber durch den Mediatrix-Server zu tunneln, ausgehebelt. Diese Funktion kann, wenn eine solche Verbindung nicht administrativ verhindert wird, positiv für die Performanz des ServiceCenter-Fensters sein, welches aus dem Mediatrix-Client aufgerufen werden kann. Dem Mediatrix-Client muss jedoch ein JDBC-Treiber zur Verfügung gestellt werden. Da aus Lizenzgründen der Treiber nicht durch ITyX signiert in das entsprechende clientlibs-Verzeichnis gelegt werden darf, muss dieser Treiber auf jeder Workstation, die das Mediatrix-Service-

Center verwendet werden soll, im Java-JRE-Verzeichnis `/lib/ext/` gespeichert werden. Siehe auch Kapitel 2.9.5 (Alternative Nutzung des ServiceCenter-Panels).

#### *defaultreplyto*

defaultreplyto=0 (default 1)

Standardmäßig wird eine „Weiterleitung mit externer Rückantwort“ über die Zeilen des E-Mail-Headers identifiziert, speziell über die Zeile „Reply to“ bzw. „References“. Wird eine „Reference“ gefunden, die zu einer Frage in Mediatrix passt, so wird diese Rückantwort als „Antwort-Fragment“ unter dieser Frage gespeichert.

Wird dieser Schalter auf 0 gesetzt, so werden diese Antworten ausschließlich über die E-Mail-ID den Mediatrix-Fragen zugeordnet. Siehe auch `extreply.ignorekonto` und `extreply.fromidonly`.

#### *deleteattachment*

deleteattachment=1 (default 1)

Attachments werden nicht mehr logisch, sondern physikalisch von der Datenbank gelöscht.

#### *disable.rtfviewer*

disable.rtfviewer=1 (default 0)

Wenn gesetzt, dann wird der interne Java-Viewer deaktiviert. Es wird der dem Betriebssystem bekannte externe Viewer für RTF-Dokumente verwendet. Die Java-lib hat auch heute noch Schwierigkeiten bei neueren RTF-Dokumenten, was wir von unserer Seite nicht beheben können! Workaround.

#### *disableFAQview*

disableFAQview=1 (default 0) Erkannte Fragestellungen (FAQ) werden in eingehenden Fragen weder blau markiert noch verlinkt.

#### *disableQuotechar*

disableQuotechar=1 (default 0)

Eine Frage, die zitiert wird, erhält normalerweise ein „>“ als Zitierzeichen vor jeder Zeile. Mit diesem Schalter entfällt dieses Zeichen.

#### *draganddrop*

draganddrop=1 (default 0)

Ist dieser Schalter aktiviert, ist es möglich Fragen aus der Mailinbox über Drag&Drop in andere Teilprojekte zu verschieben. Im Code steht übrigens eine Abfrage auf true und false, scheinbar ist das an dieser Stelle aber äquivalent.

Hinweis: Dieses Feature nicht in einer VM (Oracle Virtualbox) einsetzen. Die VM könnte abstürzen.

#### *ecat.maxresult*

ecat.maxresult=100 (default 50)

Dieser Schalter legt fest, wie viele Dokumente gleichzeitig durch den e-Kat (Externe Klassifizierung) von Contex nach Mediatix übertragen werden dürfen. Standardmäßig sind es 50 Dokumente pro Durchlauf. Ein Durchlauf wird durch den Email-Demon ausgelöst.

#### *editor.defaultfontstyle*

editor.defaultfontstyle=font-size:20pt;font-family:sansserif;font-weight:normal;

(default font-size:12pt;font-family:Arial;font-weight:bold;)

Mithilfe dieser Option kann die Standard-Schriftart im HTML-Editor des Mediatix-Clients (z.B. zum Verfassen von Antworten) umgestellt werden. Standardmäßig ist Schriftgröße 12 und „Arial“ eingestellt. Das hier eingefügte Beispiel hätte folgende Auswirkungen auf den Antwort-Editor.

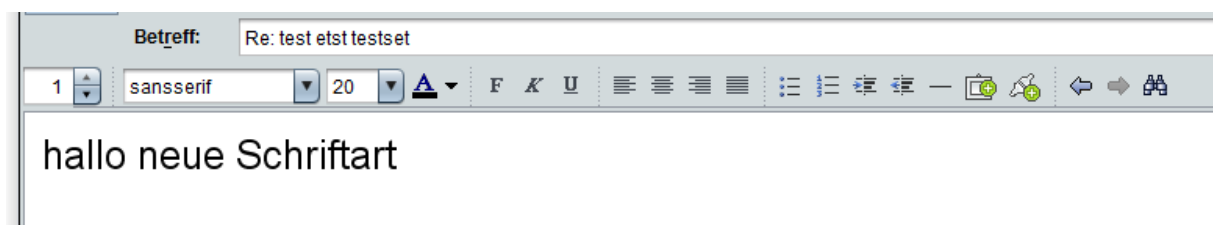


Abbildung 11: Änderung der Standard-Schriftart im Mediatix-Client

#### *emailaddresssuggest*

emailaddresssuggest=1 (default 0)

Aktiviert einen Menüpunkt, um die E-Mail-Adressvorschläge zu liefern. Da diese Funktion jedoch sehr datenbank-intensiv ist, wird sie in der Regel nicht verwendet.

#### *emaildemon.threads.count*

emaildemon.threads.count=<Zahl> (default 8)

Beispiel: emaildemon.threads.count=16

Standardmäßig holt der Email-Demon 8 E-Mail-Konten parallel ab. Falls es hier zu Verzögerungen kommen sollte, weil deutlich mehr Konten eingerichtet sind, ist es empfehlenswert, die Thread-Anzahl oder das Abhol-Intervall zu erhöhen. Diese Einstellung kann auch als Workaround dienen, wenn zu viele Threads instabil wirken (z.B. `emaildemon.threads.count=1`).

#### *encodeqp*

`encodeqp=1` (default 0)

Standardmäßig werden Emails mit dem Content-Transfer-Encoding „8bit“ über SMTP versendet, wird er auf 1 geschaltet, wird als Encoding „quoted printable“ verwendet.

#### *extreply.ignorekonto*

`extreply.ignorekonto=1` (default 0)

Standardmäßig wird das Eingangskonto der externen Antwort ignoriert. D.h. Es wird nicht überprüft, ob die Antwort auf eine „Externe Weiterleitung“ über das gleiche Konto eingegangen ist wie die ursprüngliche Kundenanfrage. Stellt man es auf 1, dann muss die externe Antwort von demselben Konto kommen wie die ursprüngliche Kundenanfrage. Siehe auch `extreply.fromidonly` und `defaultreplyto`.

#### *extreply.fromidonly*

`extreply.fromidonly=1` (default 0)

Standardmäßig werden externe Rückantworten über die References identifiziert. Stellt man diesen Schalter auf 1, so wird die Rückantwort anhand der Betreffs-Zeile identifiziert. Dieser Schalter ist direkt abhängig von `defaultreplyto=1`. Siehe auch `extreply.fromidonly` und `defaultreplyto`.

#### *filter.debug*

`filter.debug=<Zahl>` (default 0)

Beispiel: `filter.debug=1`

Ist dieser Schalter aktiviert, so werden alle PTME-Aktionen im Mediatrix-AppServer-Log ausgegeben. Diese Logausgaben sollten nur zu Diagnose-Zwecken aktiviert werden, da sie sonst das Log-File zumüllen.

#### *ignoremessageid*

`ignoremessageid=1` (default 0)

Die abzuholenden Emails, deren MessageIDs in der Datenbank schon vorhanden sind, werden beim Abholen ignoriert. Das verhindert doppelte Nachrichten im System.

Ab Mediatix v2.2.1 kann der Schalter spezifischer eingestellt werden, um die richtigen Nachrichten zu filtern. Der Schalter `ignoremessageid` ignoriert Emails, wenn sich im System bereits eine E-Mail mit identischer MessageID befindet. Diese Überprüfung findet in 2.0 und 2.1 systemweit statt, so dass eine (externe) Weiterleitung von einem Projekt in ein anderes nicht möglich ist, wenn dieser Schalter aktiviert ist. Deswegen bekommt dieser Schalter neue gültige Werte:

`off`: Deaktiviert die Erkennung identischer MessageIDs (default)

`on`: Aktiviert die Erkennung identischer MessageIDs systemweit

`system`: ist mit „on“ identisch

`project`: Aktiviert die Erkennung identischer MessageIDs projektweit<sup>10</sup>

Verhaltenskompatibilität:

Die Werte 0 und 1 werden als `off` respektive `on` interpretiert. Wird dieser Wert in der `mediatrix.properties` gefunden, so ist eine Meldung im Log (Level WARNING) auszugeben, dass dieser Wert veraltet ist.

#### *img.useexternal*

`img.useexternal=1` (default 0)

Der Schalter aktiviert den Aufruf eines Konsolenkommandos im Betriebssystem, um für Grafiken im Anhang einen externen Viewer (z.B. Paint) zu verwenden. Standardmäßig wird der interne (sehr spartanisch gehaltene) Viewer von Mediatix verwendet.

#### *jnlp.host*

`jnlp.host=12345` (kein default-Wert)

URLs, die innerhalb der JNLP erzeugt werden, wird dieser Hostname zugewiesen.

#### *jnlp.port*

`jnlp.port=1337` (kein default-Wert)

URLs, die innerhalb der JNLP erzeugt werden, wird dieser Port hinzugefügt.

#### *jnlp.use.https*

`jnlp.use.https=1` (default 0)

Dieser Schalter erzwingt die Verwendung von `https` als Schema in den generierten JNLP-Dateien.

---

<sup>10</sup> die Option „projekt“ hat denselben Effekt, ist aber DEPRECATED

*kampagnenurl*

kampagnenserverurl=<URL> (default #auskommentiert)

Beispiel: <http://127.0.0.1:8080/mediatrix/mediatrixkampagne/>

Emaillisten und Kampagnen werden in Mediatix unter Emaillisten konfiguriert. Die angegebene URL dieses Schalters ist der öffentlich verfügbare Pfad (outside global address), wo eine Bild-Datei zu finden ist, die in den erzeugten Email-Listen als Externer Inhalt angehängen wird. In der Regel hat diese Bild-Datei eine Größe von 0kb. Der Emaillisting-Demon erzeugt diese Datei jeweils einmal pro Email (Email-ID) und prüft, ob diese Datei abgerufen wird. Der Abruf dieser nicht sichtbaren Bild-Datei dient als Lesebestätigung der Email des Kunden. Blockt der Kunde externe Inhalte, so ist diese Methode natürlich nicht verlässlich. Im ServiceCenter-Vorgang sind die versendeten Emails wiederzufinden und erhalten, sobald eine Lesebestätigung vorliegt, jeweils ein Häkchen, um zu erkennen, welche der Mails angesehen wurden.

*kunde\_geschaefts\_tel\_2*

Beispiel: kunde\_geschaefts\_tel\_2=Anzahl der Haustiere (default Geschäfts Tel 2)

Standardmäßig gibt es für Kunden einen Eintrag "Geschäfts Tel 2", in der die Position innerhalb der Firma eingetragen werden kann. Falls dieses Feld mit anderen Inhalten gefüllt werden soll, kann die Anzeige innerhalb Mediatix von „Geschäfts Tel 2“ auf eine andere Bezeichnung umgeändert werden. Es sind auch Leerzeichen erlaubt. Analog dazu funktionieren auch die mediatrix.properties-Schalter kunde\_position und kunde\_internet.

*kunde\_internet*

Beispiel: kunde\_internet=Farbe des Chef-Sessels (default Internet-Adresse)

Standardmäßig gibt es für Kunden einen Eintrag "Internet-Adresse". Falls dieses Feld mit anderen Inhalten gefüllt werden soll, kann die Anzeige innerhalb Mediatix von "Internet-Adresse" auf eine andere Bezeichnung umgeändert werden. Es sind auch Leerzeichen erlaubt. Analog dazu funktionieren auch die mediatrix.properties-Schalter kunde\_position und kunde\_geschaefts\_tel\_2.

*kunde\_position*

Beispiel: kunde\_position=Anzahl der Haustiere (default Position)

Standardmäßig gibt es für Kunden einen Eintrag „Position“, in der die Position innerhalb der Firma eingetragen werden kann. Falls dieses Feld mit anderen Inhalten gefüllt werden soll, kann die Anzeige innerhalb Mediatix von „Position“ auf eine andere Bezeichnung umgeändert werden. Es sind auch Leerzeichen erlaubt. Analog dazu funktionieren auch die mediatrix.properties-Schalter kunde\_internet und kunde\_geschaefts\_tel\_2.

*language.detect.stoplenght*

language.detect.stoplength=1 (default 0)

steht der Schalter auf 1, so wird nicht die Gesamtzahl der Stopworte einer Sprache, sondern die Anzahl der unterschiedlichen gefundenen Stopworte als Basis für die Spracherkennung verwendet.

#### *language.newdetect*

language.newdetect=1 (default 0)

Schaltet die Spracherkennung auf den Stoppwort-basierten, internen Spracherkenner um.

#### *loading.content.external*

loading.content.external=0 (default 1)

Blendet den Button Externe Inhalte nachladen im Fragefenster aus, sodass der Operator keine Möglichkeit hat externe Inhalte wie Grafiken oder Scripts aus dem Internet manuell nachzuladen.

#### *loading.optimized*

loading.optimized=1 (default 0)

aktiviert das optimierte Laden, d.h. nur die Vorgänge mit noch offenen Fragen werden angezeigt. Erledigte Vorgänge können über ein Icon am unteren Ende der Liste offener Fragen sichtbar gemacht werden. Die Option lässt sich im Client pro Session abändern, beim nächsten Neustart ist wieder die globale Einstellung der mediatrix.properties aktiv.

#### *load\_db\_conf*

load\_db\_conf=0 (default 1)

Wenn der Schalter nicht gesetzt ist bzw. auf 1 steht, so werden die Parameter für Mediatrix bevorzugt aus der Datenbank geladen. Einstellungen, die in der mediatrix.properties gespeichert werden, werden dadurch unter Umständen überschrieben. Nur wenn load\_db\_conf=0 in der mediatrix.properties eingetragen ist, werden alle Schalter so wie sie in der Datei angegeben sind von Mediatrix geladen. Alternativ muss man die vorhandenen Optionen in der Datenbank löschen, damit sie erneut von der mediatrix.properties eingelesen werden können. Das SQL-Kommando dafür lautet `delete from servicecenter where id > 100`.

#### *lockdebug*

lockdebug=1 (default=0)

Dieser Schalter aktiviert, daß das Setzen und Lösen von Sperren mit in einer Datei mitprotokolliert wird.

#### *logdb*

logdb=1 (default 0)

Die Meldungen der Dienste können zusätzlich in die Datenbank geschrieben werden. Bleibt dieser Schalter auf 0, so werden ausschließlich Dateien verwendet. Aufgrund der regelmäßigen Logausgaben ist ein logging in der Datenbank nicht empfehlenswert.

#### *logeverymail*

logeverymail=1 (default 0)

Ist dieser Schalter auf 1, so wird die Tabelle report\_maileingang befüllt. Über diese Tabelle wird der Report „Maileingang“ generiert<sup>11</sup>.

#### *logForwardingAttachments*

logForwardingAttachments=1 (default 0)

Ist dieser Schalter in der mediatrix.properties gesetzt, werden weitergeleitete Attachments ins Log geschrieben.

#### *logouttime*

logouttime=360 (default 0)

Standardmäßig werden Clients niemals vom Server getrennt, es sei denn das Keepalive-Ping-Signal wird 12min lang nicht gesendet (z.B. wenn der Rechner einfach ausgeschaltet wurde). Ist der Schalter logouttime gesetzt, erhalten Operatoren, die über XX Minuten keine Aktion<sup>12</sup> durchgeführt haben, für 30 Sekunden einen Warn-Dialog, dass sie vom Server getrennt werden, sollten sie nicht den „Abbrechen“-Button drücken. Drückt der Operator diesen Button nicht, so wird er automatisch abgemeldet.

#### *logpath*

logpath=<Pfad> (z.B. ../../logs)

Legt fest, in welchem Verzeichnis die Log-Dateien gespeichert werden. Unter Windows ist unter Umständen bei mehreren Benutzern empfehlenswert einen statischen Pfad einzutragen. Standardmäßig wird logs im Mediatrix-Verzeichnis auf dem Application-Server verwendet.

#### *log.debugroll*

---

<sup>11</sup> In manchen alten erps.cfg-Einträgen (Mediatrix v1.3) steht dieser Schalter als „reporteverymail=1“, welcher aber falsch bzw. DEPRECATED ist

<sup>12</sup> Mediatrix Client erkennt auch Mausbewegungen als Aktion an. Erst wenn keinerlei Eingabegeräte verwendet werden, wird der Timer fortgesetzt



log.debugroll=<Zahl> (default 1)

Legt eine Anzahl von Logs fest, bevor sie überschrieben werden. Dieser Schalter sollte in Verbindung mit log.filesize verwendet werden. Wenn die maximale Größe einer Logdatei erreicht wird, so wird diese in %logfilename%.0 bis %logfilename.X, mit X = <Zahl>, gespeichert, bevor die ältesten Daten überschrieben werden.

#### *log.filesize*

log.filesize=<Zahl> (default infinite bzw. #Auskommentiert)

Einstellung für die maximale Größe pro Logfile (in Bytes) für Mediatix auf der Server-Seite. In der Regel wird dieser Wert jedoch durch die Einstellungen innerhalb des Tanuki-Wrappers überschrieben, welche die Ausgaben des Dienstes über die Wrapper-Funktion in ein anderes Log umleitet.

#### *log.level.archiv*

log.level.archiv=DEBUG (default INFO)

Dieser Schalter bestimmt den Log-Level des Archivierungsdienstes, welcher in die Log-Datei „archivdienst.log“ schreibt. Es stehen die Log-Level des Moduls „Log4J“ zur Verfügung (siehe <http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html>), die gängigsten sind „INFO“, „WARN“ und „DEBUG“.

#### *log.level.db*

log.level.db=DEBUG (default EXCEPTION)

Dieser Schalter bestimmt den Log-Level des Sequencers, welcher in die Log-Datei „sequencer.log“ schreibt. Es stehen die Log-Level des Moduls „Log4J“ zur Verfügung (siehe <http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html>), die gängigsten sind „INFO“, „WARN“ und „DEBUG“.

#### *log.level.email*

log.level.email=DEBUG (default INFO)

Dieser Schalter bestimmt den Log-Level des Sequencers, welcher in die Log-Datei „emailedemon.log“ schreibt. Es stehen die Log-Level des Moduls „Log4J“ zur Verfügung (siehe <http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html>), die gängigsten sind „INFO“, „WARN“ und „DEBUG“.

#### *log.level.escalation*

log.level.escalation=DEBUG (default INFO)

#### *log.level.outbound*

**\*\*to be determined\*\***

*log.level.ptme*

**\*\*to be determined\*\***

*log.level.servlet*

**\*\*to be determined\*\***

*log.keep*

**\*\*to be determined\*\***

*logtable.format*

Dieser Schalter manipuliert die Anzeige des Datums im Mitarbeiterlog.

*mail.emptyto.allowed*

mail.emptyto.allowed=true (default false)

Vorsicht wegen der leicht verdrehten Logik! Steht der Schalter auf 'true', so ist es nicht erlaubt eine Mail mit leerem Empfängerfeld abzuschicken. Steht es auf 'false', so wird die E-Mail bei leerem Empfängerfeld einfach die Adresse der Frage geschickt.

*mail.imap.auth.plain.disable*

mail.imap.auth.plain.disable=1 (default: 0)

Über diesen Schalter können Authentifizierungsmethoden (de-)aktiviert werden, dieser Schalter kontrolliert den PLAIN-Mode. Wenn Exchange 2007 IMAP-Konten verwendet werden, muss dieser Schalter auf 1 gesetzt werden.

*mail.imap.auth.ntlm.disable*

mail.imap.auth.ntlm.disable=1 (default 0)

Über diesen Schalter können Authentifizierungsmethoden (de-)aktiviert werden, dieser Schalter kontrolliert den NTLM-Mode. Wenn Exchange 2007 IMAP-Konten verwendet werden, muss dieser Schalter auf 1 gesetzt werden.

*max.forwarding.count*

Beispiel: max.forwarding.count=1000 (default 3)

Standardmäßig kann eine Email von einem Benutzer (A) höchstens 3x weitergeleitet werden. Das heißt, wenn eine Mail kommentiert in ein Teilprojekt weitergeleitet wird und ein Benutzer (B) diese Mail im neuen Teilprojekt ablehnt (Mail öffnen, im Kommentar-Feld auf ablehnen klicken), so wird die

Mail wieder in das ursprüngliche Teilprojekt verschoben. Dieser Vorgang darf insgesamt 3x wiederholt werden, bevor diese Weiterleitung verhindert wird. Es erscheint eine entsprechende Meldung für den Operator. Mit diesem Schalter kann die Anzahl von Weiterleitungen manipuliert werden, falls diese Mechanik als störend empfunden wird.

#### *maxreportsize*

maxreportsize=30 (default 10)

Dieser Schalter gibt an, wie groß ein erzeugter Report maximal sein darf. Übersteigt der Report im Falle von maxreportsize=13 die Grenze 13 Megabytes, so steht in der Datei statt des Reports eine Fehlermeldung, dass die maximal zulässige Reportgröße überschritten wurde. Die Standardeinstellung liegt bei 10 Megabytes.

#### *mediatrixmailserverurl*

mediatrixmailserverurl=<URL> (z.B. http://127.0.0.1:8080/mediatrix/mediatrixemail/)

**\*\*to be determined\*\***

#### *micolumn.n*

micolumn.5=Partner (kein default-Wert)

Mithilfe von Business-Rules ist es möglich, in der Tabelle „email“ bis zu 12 zusätzliche Extra-Felder mit individuellen Inhalten zu füllen. Dieser Parameter legt fest, mit welcher Bezeichnung das Extra-Feld in der Mail-Inbox angezeigt wird. Im o.g. Beispiel hieße dies, dass Inhalte des Feldes „extra5“ in der Mail-Inbox in der Spalte „Partner“ aufgelistet würden.

Achtung Falle: micolumn.n muss in einer zusammenhängenden Liste eingetragen werden. D.h. es muss von 1 (micolumn.1=test) angefangen werden zu zählen. Wird diese Liste von 1 bis 12 unterbrochen, so werden alle Spalten nach dieser Unterbrechung nicht mehr dargestellt. Beispiel: micolumn.1=spalte1, micolumn.2=spalte2, micolumn.4=spalte4, micolumn.5=spalte5!, „spalte4“ und „spalte5“ werden nicht mehr angezeigt, weil „micolumn.3“ fehlt.

#### *newmailsendkey*

newmailsendkey=<modifier><key> (default #auskommentiert) (z.B. alt shift s oder alt s)

Belegt die Tastenkombination STRG-S für neue Email senden neu und löst so die Doppelbelegung für neue E-Mail senden und Arbeit unterbrechen. <modifier> können alt, ctrl, meta und shift sein, <key> kann jedes Zeichen der Tastatur sein.

#### *noagentreports*

noagentreports=1 (default 0)

Dieser Schalter deaktiviert folgende Reports:

Agent Activity

Agent Status

Entire

Operator Statistics

*nouseindex*

nouseindex=1 (default 0)

Ist dieser Schalter aktiviert, so werden Datenbank-Indizes, die in der Mail-Inbox sowie in der Mail-Outbox in der Regel verwendet werden nicht forciert. Standardmäßig werden die Indizes unter allen Umständen verwendet. Dies kann unter gewissen Umständen schlechter sein. Ist dieser Schalter aktiviert, so wird die Entscheidung, ob die Indizes verwendet werden sollen oder nicht, der Datenbank überlassen. Die Wahl des Ausführungsplans ist in manchen Fällen besser als die Forcierung des Index. Wenn also Performance-Probleme innerhalb des Mailinbox-Modus auftreten und die Datenbank in Verdacht ist, kann mit diesem Schalter die Qualität der Indizes überprüft werden - läuft Mediatrix mit (nouseindex=1) schneller, so kann der Schalter aktiviert bleiben. Der ITyX-Support unterstützt hier in der Regel bei der detaillierten Analyse (<mailto:mediatrixservice@ityx.de>).

*on\_complete\_ignore\_changes*

on\_complete\_ignore\_changes=1 (default 0)

Wenn der Schalter deaktiviert ist (=0), so wird beim Erledigen einer Frage, die gerade in Bearbeitung ist und deren Antwort verändert wurde, standardmäßig gefragt, ob Änderungen gespeichert werden sollen. Diese Abfrage kann ausgeschaltet werden (=1), was jedoch zur Folge hat, dass Änderungen an der Antwort verloren gehen.

*operator.log.length*

operator.log.length=Länge (default 255)

Im Email-Antwort-Feld Details ? Log hat das Feld "Beschreibung" in Mediatrix UND in der Datenbank anstatt 255 nun 2048 Zeichen zur Verfügung. Auf diese Weise passen mehr Informationen ins Mitarbeiterlog.

Sollten mehr Zeichen für dieses Feld (Tabelle mitarbeiterlog, Feld parameter) benötigt werden (z.B. 8000), so muss der Administrator das entsprechende Feld auf varchar(8000) setzen und in der mediatrix.properties operator.log.length=8000 eintragen.

*opmodus.helpertable*

opmodus.helpertable=1 (default 0)

Bei diesem Schalter handelt es sich um eine potentielle Performanzverbesserung. Die Festlegung, welche Mediatrix-Frage einem Agenten als nächstes vorgelegt wird, wird aus den Zuständigkeiten der Teilprojekte errechnet. Dieses Ergebnis wird in einer SQL-Abfrage für jede Mediatrix-Frage verwendet. Um die Berechnung dieser Zahlen zu cachern, kann opmodus.helpertable=1 in die mediatrix.properties eingetragen werden. Die Liste der Teilprojekte wird dann für jeden Agenten individuell in einer Hilfstabelle zwischengespeichert. Diese Option kann in vielen Fällen zu einer besseren Performanz führen, es sollte aber hinreichend getestet werden. Siehe auch singlemodus.helpertable.

#### *opmodussperre*

opmodussperre=1 (default 0)

Wenn eine Frage eines Kunden zur Bearbeitung gesperrt ist, so wird im Operator-Modus keine andere Frage dieses Kunden einem Mitarbeiter vorgelegt - es sei denn der Kunde ist ein Service-Kunde<sup>13</sup>.

#### *opmodusright*

opmodusright=1 (default 0)

Bezieht sich auf das Operator-Modus-Recht. Ist der Schalter =1, so werden Mails innerhalb des Operator-Modus nur dann geladen, wenn der Agent innerhalb des Teilprojekts das Recht „Operator-Modus“ besitzt.

Ist der Schalter deaktiviert (=0), so erhält der Agent Mails ausschließlich Mails aus Teilprojekten, in denen er Zuständigkeit besitzt.

#### *opmodustable*

opmodustable=0 (default 1)

Diese Option ist nur für Oracle-Datenbanken verfügbar, da hier eine andere Methode zum Laden der nächsten Frage im Operator-Modus verwendet wird. Wenn also eine Oracle-Datenbank im Einsatz ist, ist opmodustable=0 empfohlen.

#### *passwordchange*

passwordchange=1 (default 0)

Wenn diese Option aktiviert (=1) ist, so kann jeder Operator sein Passwort über den Login-Bildschirm selbst ändern, sofern das alte Passwort noch bekannt ist. Wenn die Option ausgeschaltet ist, muss das

---

<sup>13</sup> In der Kundenadministration kann ein Kunde das Häkchen „Service-Kunde“ aktiviert werden, dies ist empfehlenswert, wenn es sich bei diesem Kunden um eine Webformular-Adresse handelt.

Passwort über einen Administrator geändert werden. Insbesondere, wenn Passwörter eine zeitliche Halbwertszeit haben, sollte diese Option mit Bedacht aus- oder eingeschaltet sein.

#### *popdebug*

popdebug=1 (default 0)

Mithilfe dieses Schalters werden zusätzliche Ausgaben im Log des Mediatrix-Emaildemon geschrieben. Sollte es Probleme beim Abholen von E-Mails geben, die im Log nicht schlüssig erklärbar sind, hilft ggf. dieser Schalter weiter. popdebug=1 sollte nicht dauerhaft aktiv sein!

#### *popproxy*

popproxy=remoteserver:110 (kein default-Wert)

Über popproxy kann ein anderer Server angegeben werden, an den alle empfangenen Mails weitergeleitet werden. Zusätzlich dazu wird pro E-Mail eine Text-Datei erzeugt, in der die Kommunikation zwischen den 2 Diensten dokumentiert wird. Die Mails werden trotzdem in die Mediatrix-Datenbank geschrieben, die Weiterleitung ist lediglich eine Kopie. Diese Funktion funktioniert nur mit dem POP3/ POP3S-Protokoll.

#### *prefetchsize*

prefetchsize=10 (default 5)

Wird der Operator-Modus verwendet, besteht die Möglichkeit über den Client-Schalter „prefetchactive“ Fragen im Hintergrund vorzuladen, um ein flüssiges Arbeiten zu garantieren. Standardmäßig werden bis zu 5 Fragen für den Agenten reserviert und in der richtigen Reihenfolge vorgelegt. Diese Warteschlange kann mit diesem Schalter beliebig verändert werden. Bedenken Sie, dass die vorgehaltenen Dokumente auch Arbeitsspeicher auf der Workstation benötigen. Die Zahl sollte also dem verfügbaren RAM, der Download-Geschwindigkeit und der Arbeitsgeschwindigkeit angepasst werden. Siehe auch prefetchactive. Prefetch (der Background-Download von Dokumenten) ist nur dann möglich, wenn das Routingmodell dies auch unterstützt. Die Mediatrix-Standard-Modelle können es, bei einem Customized-Modell muss dies explizit mit einprogrammiert werden.

#### *print.report.print\_directly*

print.report.print\_directly=0 (default 1)

Wenn =1, dann wird das Dokument direkt an den Drucker gesandt. Wenn false (=0), wird ein externes Programm zum Drucken verwendet.

#### *print.report.print\_command*

print.report.print\_command=acrord32 /p

Externes Programm (Adobe Acrobat Reader) zum Drucken von Reports im PDF-Format. Verwendbar, wenn *print.report.print\_directly=1*.

#### *propertypath*

Pfad auf die Datei properties.txt. Diese Datei enthält kundenspezifische Resource-Strings. So kann z.B. mit Start.TX12= Medi@trix Test System der Titel im Hauptfenster geändert werden.

#### *ptme.learnrealtime*

ptme.learnrealtime=0 (default 1)

Standardmäßig werden Textbausteine in Echtzeit durch die PTME gelernt. Wird diese Option auf 0 gesetzt, so wird die Zuordnung von Textbausteinen zu Emails nur dann durchgeführt, wenn die PTME trainiert wird. In der Praxis macht dies aber keinen Sinn, weil der Performance-Gewinn nur marginal ist. Diese Option wird in der Regel nur in Einzelfällen deaktiviert.

#### *ptmeserver*

ptmeserver=<IP-Adresse oder DNS> (z.B. localhost)

Server, auf dem der PTME-Dienst läuft (für die Verbindung der anderen Prozesse an den PTME-Dienst).

#### *ptmeuser1 bis ptmeuser4*

ptmeuser1=<server / ip>

ptmeuser2=<server / ip>

ptmeuser3=<server / ip>

ptmeuser4=<server / ip>

Anstatt <server / ip> können hier DNS- bzw IP-Adressen von Rechnern eingetragen werden. Mediatrix-Clients, die auf diesen Rechnern gestartet werden, sind befugt auf die PTME zuzugreifen und diese anzulernen. Es gibt (hardcoded) maximal 4 Rechner, von denen der Zugriff erfolgen darf.

#### *realtime.report.restart.availablesince*

realtime.report.restart.availablesince=0 (default 1)

Wenn dieser Wert auf 0 steht, dann wird im Echtzeitreport für den Fall, dass der Mitarbeiter gerade keine Frage bearbeitet, die Verfügbar-seit-Zeit nach einer Pause nicht neu gestartet, sondern fortlaufend gezählt.

#### *report.allmails.format*

report.allmails.format=1 (default 0))

Im Report „Einzelauswertung“ sind die Zeiten in diversen Spalten nicht human readable. Über diesen Schalter lässt sich der Report in hh/mm/ss betrachten.

#### *report.completedquestions.extracolumns*

report.completedquestions.extracolumns= 1;2;3;4;5;6;7;8;9;10;11;12 (kein Default-Wert))

Im Report „Erledigte Fragen“ sind können zusätzliche Spalten der SQL-Tabelle „email“, nämlich extra1 bis extra12, integriert werden. Welche dieser Spalten mit einbezogen werden sollen regelt dieser Schalter. Die Spalten-Nummern werden über ein Semikolon separiert und es dürfen nur die Ziffern 1-12 verwendet werden.

#### *report.delimiter*

report.delimiter=; (default ,)

Standardmäßig werden Spalten in CSV-Dateien mit dem Trennzeichen , getrennt. Microsoft trennt jedoch mit ;, weswegen dieser Schalter eingebaut wurde. Ist der Schalter gesetzt, wird das hier eingeragene Trennzeichen in den CSV-Dateien verwendet, die der Reportgenerator erzeugt.

#### *report.mailcompletion.finalcode.column*

report.mailcompletion.finalcode.column=extra[1-12] (kein Default-Wert))

Dieser Schalter bestimmt, welche Spalte, genauer, welches Feld, zum Befüllen der finalCode-Spalte benutzt werden soll. Es gibt keinen Default-Wert. Ist dieser Schalter nicht gesetzt, bleibt der Inhalt der Spalte finalCode leer.

#### *requeue.lockown*

requeue.lockown=0 (default 1)

Über den mediatrix.properties-Eintrag requeue.lockown=0 wird verhindert, dass die Auswahl des eigenen Mitarbeiters in der Wiedervorlage oder das Anwählen der Checkbox „Eigene Wiedervorlage“ die ComboBox zur Auswahl des Mitarbeiters deaktiviert.

#### *requeue\_on\_failure\_mode*

requeue\_on\_failure\_mode=1 (default 0)

- 0 : Teilprojekt erhält Wiedervorlage, wenn ein SMTP Fehler auftritt
- 1 : Mitarbeiter erhält Wiedervorlage, wenn ein SMTP Fehler auftritt

#### *search.answer.status.default*

search.answer.status.default=<string>



Wird über die Suche eine Antwort gesucht, dann kann über den o.g. Parameter ein bestimmter Status voreingestellt werden.

Optionen für den <string> sind:

ueberwacht

zwischenbescheid

fragment

erledigt

[\*search.question.status.default\*](#)

search.question.status.default=<string>

wird über die Suche eine Frage gesucht, dann kann über den o.g. Parameter ein bestimmter Status voreingestellt werden

Optionen für den <string> sind:

alle

erledigt

teilbeantwortet

extern-beantwortet

extern-weitergeleitet

weitergeleitet

wiedervorlage

neu

ueberwacht

[\*search.textobject.selectedTab.default\*](#)

search.textobject.selectedTab.default=<string> (default search)

Beim Einfügen von Textbausteinen (Textbaustein-Vorschlag) öffnet sich ein neues Fenster, in dem der erste Reiter „Suchen“ im Fokus ist. Dies lässt sich auch ändern, sodass jeweils ein anderer Reiter standardmäßig im Fokus ist.

Optionen für den <string> sind: search, FAQ, subproject, inquest, suggestion und Context

search = Suchen

subproject = Teilprojekt Textbausteine

suggestion = Textbaustein Vorschlag

FAQ = FAQ

inquest = Recherche

Contex = Contex

### *searchnames*

searchnames=1 (default 0)

Ist diese Option aktiv, so wird anhand der „Vornamen“-Tabelle, welche sich in der Mediatrix-Datenbank befindet, die Anrede eines unbekannten Kunden identifiziert. Ist der Kunde noch nicht bekannt, so wird anhand des Absender-Namens festgestellt, ob der Vorname innerhalb der Tabelle ist. Da die Tabelle nur weibliche Vornamen enthält, wird standardmäßig „Herr“ als Anrede verwendet. Nur wenn der Name des Kunden in der Vornamen-Tabelle gefunden wird, wird „Frau“ als Anrede gewählt.

### *serverdebug*

serverdebug=1 (default 0)

Detailliertere Log-Ausgaben des Mediatrix-Application-Servers, die zu Debug-Zwecken benötigt werden. Der Schalter sollte nicht dauerhaft in einer produktiven Umgebung aktiviert bleiben, da die Log-Dateien sonst nicht mehr übersichtlich bleiben.

### *servicecenter.visible*

servicecenter.visible=1 (default 0)

servicecenter.visible=0 deaktiviert die Anzeige des ServiceCenter Panel für die Verwaltung von Contex-Prozessen für den Hauptadministrator. Nur wenn der Schalter auf 1 gestellt wird, ist es für den Administrator-User standardmäßig sichtbar.

### *serverlanguage*

serverlanguage=en (default de) Dieser Schalter legt fest, in welcher Sprache die Fehlermeldungen ausgegeben werden, die ggf. vom Mediatrix-Server kommen könnten. Einstellungsmöglichkeiten sind: de, en, tr, it.

### *showsecond*

showsecond=<Zahl> (default 0)

Wenn ein Funktionsaufruf länger als <Zahl> Sekunden dauert, dann soll dieser ausgegeben werden, [\*serverdebug\*](#)

muss =1 sein. Diese Funktion ist dann einzusetzen, um langlaufende SQL-Statements zu identifizieren.

[\*singlemodus.helpertable\*](#)

singlemodus.helpertable=1 (default 0)

Bei diesem Schalter handelt es sich um eine potentielle Performanzverbesserung. Die Festlegung, welche damit sind Mediatrix-Benutzer gemeint, die das Mediatrix-Recht „Administrator“ besitzen Mediatrix-Fragen einem Agenten im Mailinbox-Modus angezeigt werden, wird aus den Zuständigkeiten der Teilprojekte errechnet. Dieses Ergebnis wird in einer SQL-Abfrage für jede Mediatrix-Frage verwendet. Um die Berechnung dieser Zahlen zu cachern, kann singlemodus.helpertable=1 in die mediatrix.properties eingetragen werden. Die Liste der Teilprojekte wird dann für jeden Agenten individuell in einer Hilfs-Tabelle zwischengespeichert. Diese Option kann in vielen Fällen zu einer besseren Performanz führen, es sollte aber hinreichend getestet werden. Siehe auch opmodus.helpertable.

[\*singlemoduslimit\*](#)

singlemoduslimit=30000 (default 15000)

Dieser Schalter überschreibt das maximale Limit von 15000 gleichzeitig anzuzeigenden Emails im Client. Diese Option ist mit Vorsicht zu genießen, da bei zu vielen angezeigten Mails der Mediatrix-Client entsprechend träge werden kann. Standardmäßig zeigt der Mediatrix-Client auch nur 7500 Mails an. Das Limit kann client-seitig bis zu dem hier eingestellten Wert genau eingestellt werden.

[\*singlemoduslimit.<ID>\*](#)

singlemoduslimit.1=45000 (default 15000)

Analog zu [\*singlemoduslimit\*](#), nur dass das Limit hier nur für einen bestimmten Mediatrix-Mitarbeiter modifiziert wird. In diesem Beispiel darf der Mitarbeiter mit der ID „1“ nun statt maximal 15000 nun 45000 Fragen auflisten. Für alle anderen Mitarbeiter gelten die globalen Einstellungen über den Schalter [\*singlemoduslimit\*](#). Das Limit kann client-seitig bis zu dem hier eingestellten Wert genau eingestellt werden.

[\*spellchecker.showsuggestion\*](#)

spellchecker.showsuggestion=1 (default 0)

Wenn aktiviert, können durch die Rechtschreibprüfung unterstrichene Wörter angeklickt werden, um Verbesserungsvorschläge zu erhalten.

[\*smtp.timeout\*](#)

`smtp.timeout=120000` (default 60000)

Definiert die Timeout-Zeit in Millisekunden, bis wann eine E-Mail versendet sein muss (Socket I/O timeout value).

#### *smtp.connection.timeout*

`smtp.connection.timeout=120000` (default 60000)

Definiert die Timeout-Zeit in Millisekunden, bis wann eine Verbindung zum SMTP aufgebaut sein muss (Socket connection timeout value).

#### *smtpdate*

`smtpdate=1` (default 0)

Gibt an, ob beim Einlesen der SMTP-Zeitstempel (aus dem received: Header-Feld) oder die Systemzeit als Eingangszeitpunkt der Nachricht verwendet werden soll.

#### *smtptrace*

`smtptrace=1` (default 0)

Erzeugt Logdateien der Kommunikation zwischen dem Tomcat und SMTP-server, siehe auch `session.setDebug = true`, dieser Schalter leistet genau dasselbe.

#### *splitsubmenu*

`splitsubmenu=50` (default 20)

Beschränkt in den Popups bei Weiterleitung auftauchenden Einträge auf die genannte Anzahl. Die Funktion wurde ab v2.0.107 auch für Unterteilprojekte (`subproject name = teilprojekt.unterteilprojekt`) angepasst. D.h. Die Weiterleitung zeigt maximal diese Anzahl Teilprojekte an, der Rest wird analog unter der Teilprojektliste über eine Kaskade verfügbar.

#### *sqllogextra*

`sqllogextra=1` (default 0)

Mithilfe dieses Kommandos können die SQL-Befehle, die durch den Mediatix-Server auf der Datenbank ausführt, in einer separaten LOG-Datei (`sql.log`) gesammelt werden. Zusätzlich dazu werden Statistiken in der Datei festgehalten. Dieser Befehl hat nur dann Wirkung, wenn gleichzeitig `serverdebug=1` geschaltet wurde. Die Log-Größe kann mithilfe `trace.sqllog.size=<Größe>` manipuliert werden. Es handelt sich hierbei um eine Debug-Funktion, sie sollte nicht dauerhaft in einer produktiven Umgebung geschaltet sein.

#### *supportemail*

supportemail=<string> (kein default-Wert)

Beispiel: supportemail=mediatrixservice@ityx.de

Wenn die Option eingeschaltet ist, so wird unter „Hilfe“ ein zusätzlicher Link eingeblendet. Klickt man da drauf, wird eine neue Email an mediatrixservice@ityx.de generiert, inkl Logs. Sollten regelmäßig Probleme (Projektphase) auftauchen, kann man hiermit komfortabel Supportmails erzeugen. Die Option ist jedoch belastend für den Applikationsserver.

#### *stats.lazy*

stats.lazy=1 (default 0)

Aus Performanzgründen kann über die Einstellung 2.9.1.3.140=1 die Berechnung der Projektstatistik geändert werden. Unter Statistiken → Projekte werden auf diese Weise nur Werte für die Teilprojekte ermittelt.

#### *stream.zip.size*

stream.zip.size=32784 (default 8196)

Dieser Schalter bestimmt, wie groß ein Objekt mindestens sein muss, damit die Kommunikation komprimiert wird. Es wird also die Größe jedes Aufrufs mit diesem Wert (in Bytes) verglichen und dynamisch entschieden, ob der Aufruf, egal in welche Richtung, zu komprimieren ist. Wenn der Wert des Schalters „0“ ist, wird jeder Aufruf komprimiert geschickt. Dieser Schalter ersetzt die mediatrix.properties-Einstellung [zipstream](#).

#### *stresstest*

stresstest=1 (default 0)

Eine Legacy 1.3 property, die für Lasttests gedacht war (Operatormode → Vorgänge öffnen, beenden etc), die unter 2.0 noch nicht eingesetzt wurde. Diese Funktion ist in der 2.0 noch nicht ausgiebig getestet und sollte deshalb unter keinen Umständen auf einem Live-System ausprobiert werden!

#### *to.clearsearchfield*

to.clearsearchfield=true (default false)

Dieser Schalter stellt sicher, dass bei der Suche nach Textbausteinen das Suchfeld immer leer bleibt und alte Suchbegriffe nicht mehr auftauchen. Standardmäßig bleiben alte Suchbegriffe eine Zeit länger im Cache.

#### *tbsearch*

tbsearch=1 (default 0))

wenn auf 1 gestellt, so ist es nicht möglich mehr als 1 parallel laufende Textbausteinsuchen zu initiieren. Die Option ist dann sinnvoll, wenn die Benutzer schnell durch die Mails klicken (z.B. Mailinbox). Bei `tbsearch=0` würden passende Textbausteinsuche-Funktionen im Hintergrund gestartet, aber nicht abgebrochen, wenn die jeweilige Mail geschlossen wird. Das führt ab einer gewissen Anzahl zu Performance-Problemen.

#### *thumbnailWidth*

`thumbnailWidth=100` (default 60))

Mit diesem Parameter die Breite von Thumbnails, das Vorschaubild, welches z.B. während der Trennung von Dokumenten in der Validierungsstation des Mediatrix-Clients verwendet wird, konfiguriert werden.

#### *thumbnailHeight*

`thumbnailHeight=120` (default 85))

Mit diesem Parameter die Höhe von Thumbnails, das Vorschaubild, welches z.B. während der Trennung von Dokumenten in der Validierungsstation des Mediatrix-Clients verwendet wird, konfiguriert werden.

#### *telefonkontakt.max*

`telefonkontakt.max=100` (default 100)

Gibt an, wie viele Suchergebnisse maximal bei der Suche im Telefonkontakt zurückgeliefert werden.

#### *trace.size*

`trace.size=1` (default 0)

Aktiviert die Funktionalität des Performance-Monitors im Mediatrix-Client. Diese Funktion sollte nur von einem Client im Netzwerk gleichzeitig verwendet werden, weil die Messung auch Leistung benötigt. Der Performance-Monitor wird über den Client im Menüpunkt Hilfe → Extras → Performancemessung aktiviert. `trace.size` sollte immer in Verbindung mit `trace.timing` verwendet werden.

#### *trace.sqllog.size*

`trace.sqllog.size=0x1000` (default 0x100000)

Dieser Schalter manipuliert die maximale Größe der Datei `sql.log`. Diese Datei wird genau dann im LOG-Verzeichnis erzeugt, wenn `serverdebug=1` und `sqllogextra=1` gleichzeitig geschaltet wurden.

#### *trace.timing*

`trace.timing=1` (default 0)

Aktiviert die Funktionalität des Performance-Monitors im Mediatrix-Client. Diese Funktion sollte nur von einem Client im Netzwerk gleichzeitig verwendet werden, weil die Messung auch Leistung benötigt. Der Performance-Monitor wird über den Client im Menüpunkt Hilfe → Extras → Performancemessung aktiviert. [trace.timing](#) sollte immer in Verbindung mit [trace.size](#) verwendet werden.

#### [viewpermission.ignore](#)

viewpermission.ignore=1 (default 0)

In der Kundenhistorie sind in 2.0 nur noch Mails von Teilprojekten anzeigbar, für die man selber zuständig ist. Das ist eine Einschränkung, die in den allermeisten Fällen die effektive Arbeit behindert. Empfohlener Wert für alle Neuinstallationen ist 1.

#### [welcomepanel.implementor.1, 2 und 3](#)

welcomepanel.implementor.1=empty (default default)

Dieser Schalter blendet die rechten Elemente des Startbildschirms des Mediatrix-Clients ein (=default) oder aus (=empty). Dazu gehört die Liste die Statistiken, die eigenen Wiedervorlagen sowie Benutzer-Informationen.

#### [windowseventlog](#)

windowseventlog=1 (default 0)

Die Datei ntlog.dll wird benötigt. Die Funktion schreibt Ereignisse des Mediatrix-Applicaton-Servers in das Windows-Eventlog.

#### [wv.verzoegerung](#)

wv.verzoegerung=XX (default 0)

Wenn  $XX > 0$ , dann wird die Wiedervorlage um mindestens XX minuten in die Zukunft gelegt. Ein sofortiges Wiedervorlegen verzögert sich also um mindestens XX Minuten. Beispiel: wv.verzoegerung=60. Eine Wiedervorlage 2 Stunden in die Zukunft wird nach 2 Stunden wiedervorgelegt, eine Wiedervorlage 30min in die Zukunft wird nach 60min wiedervorgelegt.

### 4.4.3 Single-Sign-On

Mediatrix unterstützt den direkten Login (Auto-Login) über die Benutzer-Login-Daten des Betriebssystems. Voraussetzung dafür ist ein Domain-Controller in Form eines Active Directories (Kerberos). Ist es korrekt konfiguriert, so werden beim Start des Mediatrix-Clients die Daten des gerade eingeloggtten Benutzers vom AD übernommen. Beim Start der Anwendung wird jetzt zunächst der am Windows-System angemeldete „Principal“ anhand des Kerberos-Ticket-Cache ermittelt. Wird jetzt in

Mediatrix ein Benutzer mit passendem Anmeldenamen (Benutzername ohne folgende @DOMAIN Angaben) identifiziert, wird dieser sofort angemeldet.

Beispiel: Ist der Benutzer angemeldet als ITYX.DE\meinBenutzer und hat das Kennwort „ityx“, so muss der korrespondierende Mediatrix-Benutzer „meinBenutzer“ und das Mediatrix-Passwort „ityx“ sein. Für den Schalter sso.domain muss in diesem Beispiel „ITYX.DE“ konfiguriert sein. Für den Abgleich mit dem Active Directory muss zudem der entsprechende Domain Controller hinterlegt sein.

#### *sso.active*

sso.active=1 (default 0)

Aktiviert die Single-Sign-On-Unterstützung für Mediatrix. Wenn PCs, auf denen Mediatrix-Client gestartet wird, in ein Active Directory (Kerberos) an einem Domain-Controller angebunden sind, so ist es möglich Mediatrix ohne zusätzlichen Login zu starten. Voraussetzung dafür ist, dass die Login-Namen und Kennwörter identisch zu den Windows-Logon-Daten sind.

#### *sso.domain*

sso.domain=ITYX.de (kein Default-Wert)

Bezeichnet die Anmelde-Domäne für PCs, auf denen Mediatrix läuft. Für Single-Sign-On kann pro Mediatrix-Unit nur eine Domain eingestellt werden. Alle anderen Clients müssen sich dann regulär anmelden.

#### *sso.domain\_pdc*

sso.domain\_pdc=dc.ityx.de (kein Default-Wert)

Hier muss die vom Mediatrix-Server erreichbare Adresse des Domain-Controllers eingetragen werden.

#### *sso.checkdomain*

sso.checkdomain=1 (default 0)

Dieser Schalter ist immer dann zu aktivieren, wenn der Domainname ein Teil des Login-Namens ist. Z.B. Domain=ityx.de, Login-Name=ityx.de\_user1 oder user1@ityx.de.

#### *sso.fullprincipal*

sso.fullprincipal=1 (default 0)

#### *sso.configPath*

sso.configPath=1 (default 0)



#### 4.4.4 Alternative FunctionChecker-Erweiterung

*Diese alternative Semaphoren-Technik sollte nur dann eingesetzt werden, wenn es unter Mediatix zu enormen Performance-Problemen kommen sollte.*

Generell wird der Zugriff von Mediatix auf die Datenbank über sogenannte „Synchronized Code-Blocks“ verwaltet. D.h. wenn eine Schreiboperation auf einen bestimmten Datensatz durchgeführt werden soll, so wird die Ressource exklusiv für den Programm-Thread reserviert, der sie gerade benötigt. Alle anderen Threads, die dieselben Daten verändern müssen, werden auf eine Warteposition gesetzt (Stichwort: Semaphoren).

Es kommt jedoch vor, dass die Datenbank keine Rückmeldung sendet und diese Semaphore (=Synchronized Block) nicht mehr für andere Threads geöffnet wird, was zu Systemstillständen führt. Zu diesem Zweck wurde eine Codeänderung erstellt, die sich mit den folgenden Schaltern aktivieren lässt.

##### [blocking.functionchecker.maxsemaphores](#)

`blocking.functionchecker.maxsemaphores=4` (default 1)

Dieser Schalter erhöht die maximale Anzahl von standardmäßig einem Thread pro „Synchronized Block“ auf eine beliebige Anzahl. Bisher hat sich ein maximaler Wert von 4 Threads bewährt. Im Falle von „non-blocking“ Semaphore-Techniken (siehe [bean.functionchecker](#)) wird gewartet, bis weitere Threads den Synchronized-Block betreten dürfen. Die Semaphore wird maximal für den Zeitraum, der über [blocking.functionchecker.maxwaittime](#) konfiguriert wird, gesperrt. Nach diesem Zeitraum darf ein weiterer Thread auf den Synchronized-Block zu

greifen. Sobald die in diesem Falle 4 Threads im Synchronized-Code-Block aktiv arbeiten, werden andere

Threads warten (Thread-Action = WAIT).

##### [blocking.functionchecker.maxwaittime](#)

`blocking.functionchecker.maxwaittime=4000` (default 2000)

Im Falle von „non-blocking“ Semaphore-Techniken (siehe [bean.functionchecker](#)) wird für die angegebene Zeit in Millisekunden gewartet, bis weitere Threads eine Semaphore betreten dürfen.

##### [bean.functionchecker](#)

`bean.functionchecker=de.ityx.check.RegulatedNonBlockingOpenDoorsFunctionChecker` (kein default-Wert)

Über diesen Schalter wird eine alternative Semaphoren-Technik aktiviert. Wie bereits erwähnt sollte diese Technik nur zum Einsatz kommen, wenn die Standard-Technik zu Performance-Problemen führen sollte.

Optionen:

- `de.ityx.check.BlockingSemaphoreFunctionChecker` (experimentell)
- `de.ityx.check.NonBlockingSemaphoreFunctionChecker` (experimentell)
- `de.ityx.check.RegulatedNonBlockingOpenDoorsFunctionChecker` (unter Oracle 11g bewährt)
- `de.ityx.check.RegulatedSemaphoreFunctionChecker` (experimentell)

#### *bean.transactionHandler*

`bean.transactionHandler=de.ityx.check.AllRepeatableLimitedTransactionHandler` (kein Default-Wert)  
Sollte es zu einem Deadlock kommen, kann über dieses Bean konfiguriert werden, ob die blockierten Transaktionen wiederholt werden sollen. Wenn dieses Java-Bean geladen ist, werden alle Transaktionen bis auf „E-Mail senden“ wiederholt. Wird diese Property nicht in die Datei `mediatrix.properties` eingetragen, so wird keine „abgeschlossene“ Transaktion wiederholt.

#### *bean.syncfactory*

`bean.syncfactory=de.ityx.common.sync.SyncWithExclusionsByPropertyFile` (kein Default-Wert)

---

*Diese Property keinen Default-Wert, es kommt die „DefaultSyncFactory“ aus dem Mediatrix-Core zum Einsatz, wenn `bean.syncfactory` nicht konfiguriert wird. Sollte es hier zu Problemen kommen (und nur dann!), kann über `bean.syncfactory` ein alternatives Deadlock-Handling aktiviert werden.*

---

Optionen:

- `de.ityx.common.sync.SyncWithExclusionsByPropertyFile`
- `de.ityx.check.DeadLockSync`

`de.ityx.check.DeadLockSync` synchronisiert nichts! Sofern die Datenbank das Transaction-Handling selbständig durchführt (dies ist bei Oracle der Fall), so wird durch diese Synchronisationsmechanik eine maximale Parallelität gewährleistet. Unter anderen Datenbankmanagement-Systemen ist diese Option noch nicht erprobt.

Wird `de.ityx.check.SyncWithExclusionsByPropertyFile` eingestellt, ist eine neue Datei zu erstellen, welche im „Home“-Verzeichnis zu speichern ist (in der Regel ist dies `mediatrix/conf/server/`). Name der Datei: `sync.properties`; hier wird ein Regelwerk hinterlegt, welche Methoden nicht synchronisiert werden sollen. Auf diese Weise können bestimmte Kern-Funktionen von Mediatrix gänzlich an den Semaphoren vorbeigeführt werden. Diese Funktion birgt jedoch ein gewisses Risiko und sollte vor dem Einsatz in einer produktiven Umgebung getestet werden.

#### **Syntax**

SyncPoint::[Methode]::[Parameter-Typen]

### Beispiel

```
# Fragen: Abspeichern nicht synchronisieren
SyncPoint::store:Connection_Question_Boolean_Long_

SyncPoint::store:Connection_Question_

# Antworten: Abspeichern nicht synchronisieren
SyncPoint::store:Connection_Answer_

# Vorgänge: Abspeichern nicht synchronisieren
SyncPoint::store:Connection_Case_

# Kunden: Abspeichern nicht synchronisieren
SyncPoint::store:Connection_Customer_
```

#### 4.4.5 c3p0-Connection-Pool-Einstellungen

Für die Verbindungsverwaltung zwischen Server und Datenbank wird „C3P0“ verwendet. Dieses Connection-Pooling kann über die folgenden mediatrix.properties-Schalter verändert werden. In den meisten Fällen sollten die Standard-Werte jedoch ausreichend sein. Mehr Informationen gibt es unter <http://www.mchange.com/projects/c3p0/>.

##### *c3p0.acquireIncrement*

c3p0.acquireIncrement=2 (default 3)

Legt fest, wie viele Verbindungsversuche gleichzeitig erfolgen sollen, wenn der Connectionpool ausgeschöpft ist. Siehe auch [c3p0.acquireRetryAttempts](#), [c3p0.acquireRetryDelay](#) und [c3p0.breakAfterAcquireFailure](#).

##### *c3p0.acquireRetryAttempts*

c3p0.acquireRetryAttempts=45 (default 30)

Definiert, wie oft c3p0 einen neuen Verbindungsversuch zur Datenbank unternehmen soll, bevor der Mechanismus aufgibt. Ist dieser Wert  $\leq 0$ , so wird unendlich oft versucht die Verbindung wiederherzustellen.

Siehe auch [c3p0.acquireIncrement](#), [c3p0.acquireRetryDelay](#) und [c3p0.breakAfterAcquireFailure](#).

##### *c3p0.acquireRetryDelay*

c3p0.acquireRetryDelay=2000 (default 1000)

Angabe in Millisekunden. Der Schalter legt die Zeit fest, die C3P0 wartet, bis ein neuer Verbindungsversuch unternommen wird. Siehe auch [c3p0.acquireRetryAttempts](#), [c3p0.acquireIncrement](#) und [c3p0.breakAfterAcquireFailure](#).

#### *c3p0.autoCommitOnClose*

c3p0.autoCommitOnClose=true (default false)

Die JDBC-Spezifikation ist penetrant still in Bezug auf nicht-aufgelöste und offene Transaktionen während eines Schließens einer Verbindung. Das Standardverfahren bei C3P0 ist ein Rollback aller nicht übertragenen und offenen Aufträge. Stellt man c3p0.autoCommitOnClose=true ein, so werden offene Aufträge automatisch „committed“, statt die Transaktion teilweise zurückzurollen.

#### *c3p0.automaticTestTable*

c3p0.automaticTestTable=c3p0\_test\_table (default null)

Wenn eingestellt, wird C3P0 eine leere Tabelle gleichen Namens - in diesem Beispiel c3p0\_test\_table - erstellen und SQL-Queries auf diese Tabelle ausführen, um die Verbindung zu testen. Dieser Schalter übertrumpft [c3p0.preferredTestQuery](#), d.h. wenn [c3p0.automaticTestTable](#) gesetzt ist, werden alle Angaben für [c3p0.preferredTestQuery](#) ignoriert.

Diese Tabelle sollte für keine anderen Zwecke als den Test der Verbindung verwendet werden.

#### *c3p0.breakAfterAcquireFailure*

c3p0.breakAfterAcquireFailure=true (default false)

Wenn true, dann wird eine Datenquelle im Connection Pool die Verbindung permanent schließen, wenn die Verbindung nach der maximalen Anzahl von c3p0.acquireRetryAttempts nicht aufgebaut werden konnte. Wenn der Schalter auf false gestellt wird, so werden alle Programm-Threads, die auf eine Verbindung warten, eine Exception werfen - die Datenquelle bleibt jedoch intakt. Siehe auch [c3p0.acquireRetryAttempts](#), [c3p0.acquireRetryDelay](#) und [c3p0.acquireIncrement](#).

#### *c3p0.checkoutTimeout*

c3p0.checkoutTimeout=2000 (default 0)

Millisekunden. Wenn der Connection Pool ausgeschöpft ist, ist dies die Anzahl an Millisekunden, die gewartet wird, bis der Verbindungsversuch mit einer SQL-Exception abgebrochen wird. Wird der Wert auf 0 gesetzt, wird der Aufruf von getConnection() unbegrenzt lange auf WAIT gehalten, was dazu führen kann, dass ein Client „hängt“, sich aber wieder fangen kann, wenn wieder Verbindungen übrig sind.

#### *c3p0.debugUnreturnedConnectionStackTraces*

c3p0.debugUnreturnedConnectionStackTraces=true (default false)

Wenn true, und wenn [c3p0.unreturnedConnectionTimeout](#) einen positiven Wert hat, dann wird der Connection-Pool eine Exception für jeden Connection-Checkout speichern und ausgeben, wenn alle nicht-zurückgegebenen Check-Out-Verbindungen einen Timeout erfahren. Dieser Schalter kann dazu verwendet werden Mediatix-Applikationen zu debuggen, falls in einer Komponente ein DB-Connection-Leak vermutet wird (Bedeutet, dass immer mehr Datenbankverbindungen aufgebaut werden, die nicht geschlossen werden). Wird ein solches Verhalten nicht beobachtet, sollte dieser Schalter unbedingt auf false stehen bleiben, da die Datenbankperformanz darunter leiden wird. Siehe auch [c3p0.unreturnedConnectionTimeout](#).

#### [c3p0.idleConnectionTestPeriod](#)

c3p0.idleConnectionTestPeriod= 30 (default 0)

Wenn c3p0.idleConnectionTestPeriod > 0 ist, dann wird der Connection-Pool alle untätigen und unchecked-out Verbindungen alle XX Sekunden überprüfen.

#### [c3p0.initialPoolSize](#)

c3p0.initialPoolSize=80 (default 3)

Die Anzahl der Datenbankverbindungen im Connection-Pool, die die Applikation zu Beginn versuchen wird zu akquirieren. Diese Zahl sollte zwischen [c3p0.minPoolSize](#) und [c3p0.maxPoolSize](#) liegen.

#### [c3p0.maxConnectionAge](#)

c3p0.maxConnectionAge=28800 (default 0)

Sekunden. Im eigentlichen Sinne ist [c3p0.maxConnectionAge](#) die „Time-to-live“. Eine Verbindung, die älter ist als [c3p0.maxConnectionAge](#), wird zerstört und aus dem Connection-Pool gelöscht. Dies unterscheidet den Schalter c3p0.maxConnectionAge von [c3p0.maxIdleTime](#) - es wird das absolute Alter der Verbindung betrachtet. D.h. auch aktive Verbindungen, die das definierte Alter erreichen, werden beendet. Standardmäßig wird kein maximales Alter vorgeschrieben (=0).

#### [c3p0.maxIdleTime](#)

c3p0.maxIdleTime=14400 (default 0)

Anzahl von Sekunden, die eine Datenbankverbindung ungenutzt (=idle) im Connection-Pool verbleiben darf, bevor sie weggeworfen wird. c3p0.maxIdleTime=0 bedeutet, dass die Idle-Verbindung niemals abläuft. Dieser Wert sollte stets unter dem Datenbank-Gegenstück liegen, um Seiteneffekte zu vermeiden. Unter MySQL ist die Einstellung in der Datei my.ini `wait_timeout = 28800`

#### [c3p0.maxPoolSize](#)

c3p0.maxPoolSize=80 (default 15)

Maximale Anzahl von Verbindungen, die in einem Connection-Pool zur Verfügung stellen muss. Siehe auch [c3p0.minPoolSize](#) und [c3p0.initialPoolSize](#).

#### *c3p0.minPoolSize*

c3p0.minPoolSize=10 (default 3)

Minimale Anzahl von Verbindungen, die in einem Connection-Pool zur Verfügung stellen muss. Siehe auch [c3p0.maxPoolSize](#) und [c3p0.initialPoolSize](#).

#### *c3p0.numHelperThreads*

c3p0.numHelperThreads=5 (default 3)

C3P0 arbeitet „sehr“ asynchron. Das führt dazu, dass langsame JDBC-Operationen generell von sogenannten Helper-Threads abgearbeitet werden, die keine „contended locks“ auf der Datenbank halten. Eine Erhöhung von [c3p0.numHelperThreads](#) kann signifikant höhere Performanz einbringen, da mehrere Operationen gleichzeitig ablaufen können.

#### *c3p0.preferredTestQuery*

c3p0.preferredTestQuery=Select 1 from DUAL (kein default-Wert)

Definiert die Query, die für alle Verbindungstests ausgeführt wird. Die Query sollte sehr schnell zu einem Ergebnis führen, große Join-Operationen für einen Test sind nicht empfohlen. Die Tabelle, auf der die Query ausgeführt wird, muss existieren (also in diesem Beispiel die Tabelle DUAL). Diese Option wird ignoriert, wenn c3p0.automaticTestTable auf `true` eingestellt wird.

#### *c3p0.propertyCycle*

c3p0.propertyCycle=10 (default 0)

Die Maximale Anzahl an Sekunden, bevor die „user configuration constraints“ erzwungen werden. Dazu gehören folgende Einstellungen: [c3p0.maxConnectionAge](#), [c3p0.maxIdleTime](#) und [c3p0.unreturnedConnectionTimeout](#). C3P0 prüft in Zeitintervallen, wann diese Einstellungen überprüft werden - c3p0.propertyCycle definiert dieses Zeitintervall. Der default-Wert 0 bedeutet „automatisch“, d.h. ein sinnvoller Prüfwert wird von der Applikation selbst ermittelt.

#### *c3p0.unreturnedConnectionTimeout*

c3p0.unreturnedConnectionTimeout=60 (default 0)

Sekunden. Wenn eingestellt und wenn Mediatrix (Client) eine Verbindung belegt, diese aber nicht in den Connection-Pool zurückgibt, dann wird nach dieser definierten Zeit die Verbindung gnadenlos zerstört. Dies erlaubt Applikationen, die ein „Connection Leak“ (sich anhäufende, ungenutzte Verbindungen) produzieren, weiterhin stabil laufen zu lassen - was eigentlich eine Schande ist. Sollte

sich also jemals ein Bug in Mediatrix oder im Customizing einschleichen, der ein solches Verhalten provoziert, ist dieser Schalter eine potentielle Notlösung.

#### 4.5 Mediatrix Authentifizierung

Ort	mediatrix/conf/mediatrix.properties und krb5.conf (optional)
Notwendige Rechte	root / Administrator (Server)



Abbildung 12: Mediatrix-Login

Der Mediatrix Client bietet drei Methoden an, um sich während des Logins zu authentifizieren.

- Nativer Login (herkömmlicher Login, gesteuert über die Benutzeradministration von Mediatrix)
- Kerberos 5
- Windows-Authentifizierung

Die Mediatrix-Authentifizierung ist ein klassischer Single Sign On Mechanismus. Der aktuell verwendete Windows-Login des Benutzers wird für die Authentifizierung in Mediatrix mitverwendet, er muss lediglich „login“ drücken. Das Kennword muss nicht erneut eingegeben werden. Bei Kerberos

5 können ebenfalls Windows-Login-Daten für Mediatrix verwendet werden, auch wenn die aktuelle Windows-Session einem anderen Benutzer gehört. Mediatrix implementiert hierfür die JAAS-API<sup>14</sup>.

---

*Technisch gibt es kaum Unterschiede zwischen Kerberos und Windows-Authentifizierung. Microsoft nutzt seit Windows 2000 das Kerberos-Protokoll.*

---

Bei einem Login über Kerberos5 oder Windows-Authentifizierung wird lediglich Username + Password mit Domaincontroller oder Kerberos-Datenbasis verglichen. Die Rechte werden weiterhin über die Benutzeradministration von Mediatrix vergeben, z.B. über Profile, die in Mediatrix zugeordnet bzw. zugewiesen werden können. Eine Profil-Zuordnung, die z.B. über AD-Gruppen auf die Benutzer „gemappt“ werden, ist erst für eine spätere Version von Mediatrix geplant. Die Workstation erhält bei erfolgreichem Login über Windows-Authentifizierung oder Kerberos ein sogenanntes „TGT (Ticket Granting Ticket)“.

Die Grundvoraussetzungen für einen Login sind

- Benutzer/Kennwort Kombination muss valide sein
- Der Windows AD Benutzer, der für den Windows-Login verwendet wurde, muss ebenfalls in Mediatrix einen gleichnamigen Account besitzen. Die Kennwörter müssen jedoch nicht übereinstimmen

Beispiel:

	Windows User	Mediatrix User
User Name	Ityxuser	Ityxuser
Password	Password123123123	Ityx
Domain	ITYX.DE	n/a

- Wenn Single Sign On konfiguriert ist, kann ein Benutzer sich folgendermaßen einloggen
  - Windows Authentication to login without credentials (Current Windows Login is used)
  - Kerberos Login by using Windows User + Windows Password
  - Mediatrix Login by using Mediatrix User + Mediatrix Password

#### 4.5.1 Unterschied zwischen den Authentifizierungsmethoden

Technisch ähneln sich diese zwei Methoden „Kerberos“ und „Windows-Authentication“ sehr. Ohne zu sehr ins Detail zu gehen ist für den Anwender folgender Unterschied relevant.

---

<sup>14</sup> Oracle-Beschreibung: <http://www.oracle.com/technetwork/java/javase/jaas/index.html>

Wikipedia-Beschreibung: [http://en.wikipedia.org/wiki/Java\\_Authentication\\_and\\_Authorization\\_Service](http://en.wikipedia.org/wiki/Java_Authentication_and_Authorization_Service)



Methode	Beschreibung
<b>Nativer Login</b>	Es müssen Benutzer und Passwort eingetragen werden, die in Mediatix hinterlegt wurden. Benutzer und Passwort müssen nicht im Windows-AD existieren, sie sind frei wählbar und können auch unabhängig vom Windows-AD geändert, deaktiviert oder hinzugefügt werden. Dies ist die Standard-Login-Variante
<b>Kerberos5-Login</b>	Es wird das Kerberos-Protokoll verwendet. D.h. es muss serverseitig (auf Windows-Seite bzw. Linux-Seite sowie auf Mediatix-Seite) vollständig konfiguriert sein. Die eingegebenen Credentials werden gegen ein Active Directory (Windows) oder einen Kerberos5-Server (Linux) geprüft. Zeitgleich muss der eingetragene Benutzer im Active Directory auch einen gleichnamigen Mediatix-Account haben, damit ein Login möglich ist (z.B. AD-Benutzer heißt „ITYX.DE\meinBenutzer“, dann muss in Mediatix ein Account namens „meinBenutzer“ existieren, sonst ist kein Login möglich.
<b>Windows Authentication Login</b>	Auch hier wird das Kerberos-Protokoll verwendet. Somit ist ebenfalls eine vollständige Konfiguration vorausgesetzt. Die Credentials werden vom Login des Betriebssystems übernommen. Für einen erfolgreichen SingleSignOn ist es notwendig, dass in Mediatix ein gleichnamiger Benutzer existiert, mit dem man sich am Betriebssystem angemeldet hat (z.B. AD-Benutzer heißt „ITYX.DE\meinBenutzer“, dann muss in Mediatix ein Account namens „meinBenutzer“ existieren, sonst ist kein Login möglich. Diese Login-Variante funktioniert nur unter Windows, andere Betriebssysteme müssen den Kerberos-Login verwenden.

Wenn sich Mediatix gegen ein Active Directory authentifiziert, dann sind immer sowohl Kerberos- als auch Windows Login möglich. Die Login-Methoden sind identisch, nur dass bei Windows der aktuelle Windows-Login forciert wird, und bei Kerberos ein beliebiger Windows-Login eingegeben werden kann.

Voraussetzung für alle drei Authentifizierungsmethoden ist ein gleichnamiger Mediatix-Account. Wenn die Logins der Mediatix-Accounts unterschiedlich zum Windows/Kerberos-Login sind, ist der Login ausschließlich über die herkömmliche Variante, Native Login, möglich.

#### 4.5.2 Quick Guide

- Loggen Sie auf dem Windows Server ein, auf dem sich das Active Directory befindet
- Erstellen Sie einen Benutzer in Ihrem AD, über den die Benutzer zukünftig ihre AD-Daten zur Authentifizierung senden. Der Benutzer sollte in der Lage sein sich in Domänen-PCs einzuloggen und das Passwort sollte dauerhaft gültig sein. Dieser Nutzer wird für Mediatix-Server genutzt und nicht an Agenten weitergegeben
- Öffnen Sie eine Kommandozeile (cmd)
- Erstellen Sie ein KeyTab-File über das folgende Kommando

- Die Kommandos sind Linux- und Windows-Server-Befehle, klist und kinit sind jedoch auch in JavaSDK-Installationen verfügbar

```
ktpass -princ <username>/<active directory address>@<DOMAIN> -pass
<userpassword> -mapuser <username>@<DOMAIN> -out C:\keytab_mediatrix -ptype
KRB5_NT_PRINCIPAL -crypto all
```

### Beispiel

```
ktpass -princ xservice/dc1.ityx.de@ITYX.DE -pass ityx -mapuser
xservice@ITYX.DE -out C:\keytab_mediatrix -ptype KRB5_NT_PRINCIPAL -crypto
all
```

- Beachten Sie, dass alle Namen und Domänen absolute Beachtung bei Groß und Kleinschreibung benötigen
- Testen Sie ob ein Kerberos Ticket mit dem KeyTab file erzeugt werden kann
  - o Auflisten der Eigenschaften des Keytab Files

```
klist -k -t -e c:\keytab_mediatrix
```

- o Erstellen eines Kerberos Tickets über das KeyTab File

```
kinit -k -t c:\keytab_mediatrix <username>/<active directory address>@<DOMAIN>
```

### Beispiel

```
kinit -k -t c:\keytab_mediatrix xservice/dc1.ityx.de@ITYX.DE
```

- o Ticket betrachten

```
klist
```

- o Ticket entfernen, es wird nach dem Test nicht mehr gebraucht

```
kdestroy
```

---

*Wenn kein Ticket erzeugt werden konnte, wurde in einem der Kommandos ein Fehler begangen. Dieser Test muss erfolgreich sein, sonst wird SSO-Login in Mediatrix höchstwahrscheinlich nicht funktionieren.*

---

- Kopiere das erstellte KeyTab-File auf den Mediatrix-Server, z.B. unter mediatrix/conf/
- Editieren Sie die Datei mediatrix.properties und fügen Sie diese Einträge hinzu (editieren Sie diese passend zu Ihren ktpass-Parametern)
  - o authentication.methods=windows,kerberos,mediatrix
  - o authentication.serviceprincipal=xservice
  - o authentication.realm=ITYX.DE
  - o authentication.keydistribution=dc.ityx.de
  - o authentication.keytabfile=c:/mediatrix/conf/keytab\_mediatrix

- Beachten Sie, dass auch hier alle Namen und Domänen absolute Beachtung bei Groß und Kleinschreibung benötigen
- Installieren Sie das Java JCE package auf dem Mediatrix-Rechner
  - o <http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html> (Java 7)
  - o <http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html> (Java 8)
- Installieren Sie dasselbe Package auf allen Client-Workstation, wo Mediatrix-Client verwendet werden wird
- Restarten Sie den Mediatrix-Application-Dienst
- Testen Sie den Mediatrix-Client-Login mit "Windows Authentication"

---

*SingleSignOn Login ist nicht möglich, wenn der Benutzer über locale oder domänenweite Administrator-Rechte verfügt. Nutzen Sie stattdessen Kerberos-Login.*

---

## 5 Systembetrieb

Dieses Kapitel befasst sich mit dem Betrieb und der Konfiguration der Software, die nicht im Mediatrix-Client vorgenommen werden können, sowie der unterliegenden Systeme.

### 5.1 Tomcat Tuning

Der Tomcat-Applikationsserver ist die Grundlage für alle Mediatrix-Dienste, die mitgeliefert werden. In Mediatrix wird Tomcat v8.0 verwendet. Je nach Nutzung der Mediatrix-Software sollten unter Umständen Parameter innerhalb des Applikationsservers angepasst werden, um die Performanz und Stabilität zu gewährleisten.

Es gilt zwischen Tomcat-spezifischen Problem und Mediatrix-Problemen zu unterscheiden. Viele Connectionund Memory-Probleme können über mediatrix.properties oder die Tanuki Wrapper Konfigurationsdateien bereits optimiert oder gelöst werden, auch applikationsspezifisch (z.B. verfügbarer RAM für den Mediatrix AppServer 4GB, für den Mediatrix Email-Demon jedoch nur 2GB).

#### *HTTP Connector Parameter*

Die Datei, die für zentrale Modifikationen in Frage kommt, ist /mediatrix-context/tomcat/conf/server.xml, die meisten Parameter werden im XML-Tag „<Connector />“ eingefügt. Auswirkungen von Veränderungen an der server.xml sollten auf einem Testsystem geprüft werden, bevor sie produktiv eingesetzt werden!

Hier die wichtigsten Parameter. Es handelt sich um die Parameter, die durch den Support am häufigsten angepasst wurden.

#### *connectionTimeout*

(default 60000)

Abhängig von der Bandbreite zwischen Client und Server sowie der Größe der zu verarbeitenden Dokumente kann es notwendig sein diesen Wert von 20 Sekunden zu erhöhen, da es sonst zu SocketRead() Timeouts kommt. Werden CONTEX-Dokumente nach Mediatix und umgekehrt geschickt, so kann es vorkommen, dass ein 50MB-Dokument bei wenig Bandbreite länger als 20sek unterwegs ist (Stichwort Request & Accept). In diesem Falle muss dieser Parameter erhöht werden, z.B. 40000 (=40sek).

#### *port*

(default 8080)

Ändert den Tomcat-Port, zu dem verbunden wird. Wird hier port=80 eingetragen, so sind die Mediatix-Applikationen über <http://localhost:80/mediatrix/jnlp> erreichbar.

#### *maxThreads*

(default 200)

Ändert die maximale Anzahl der gleichzeitig laufenden Threads (= Verbindungen) zum Tomcat-Server. Wird kein Connectionpool verwendet, muss dieser Wert zwangsweise höher eingestellt sein, da jeder aktive Mediatix- Client zwischen 1-2 offene Threads benötigt. Bei mehr als 100 gleichzeitig aktiven Clients kommt es sonst zu Engpässen. Die Erhöhung dieses Werts bedeutet jedoch auch, dass der Server mehr Arbeitsspeicher im laufenden Betrieb und mehr Zeit für einen Kaltstart benötigt.

#### *Weitere CONNECTOR-Parameter*

Eine komplette Liste der Einstellungsmöglichkeiten, die für alle Connector-Typen verwendet werden können, befindet sich im Anhang. Diese sollten nur mit Bedacht und unter Absprache mit dem ITyX-Support eingesetzt werden.

Mediatix verfügt für diverse dieser Einstellungen native Optionen, die in die mediatrix.properties einzutragen sind. Nach Möglichkeit sollten diese nicht doppelt eingefügt werden, da es sonst zu Seiteneffekten kommen kann. Aufgeführte Einstellungen sollten nur mit Bedacht und unter Absprache mit dem ITyX-Support eingesetzt werden.

In der Datei server.xml, die mit Mediatix ausgeliefert wird, sind 20000ms als connectionTimeout eingetragen

#### *Auswahl des Connectors*

Apache Tomcat stellt folgende Implementationen zur Verfügung:

- Java-based HTTP/1.1 (Coyote) Connector
- Java-based High performance NIO HTTP Connector
- Native code-optimized APR HTTP Connector

Die Voreinstellung, HTTP/1.1, ist in der Regel bewährt und stabil. Der Native code-optimized APR http Connector ist ein Addon des HTTP-Connectors, der über eine DLL zusätzlich geladen wird (siehe unten), nach Möglichkeit sollte diese immer geladen werden. Die NIO HTTP Connection ist eine sehr neue Connector-Variante des Tomcat-Servers, die jedoch erst 1 Jahr erprobt ist. Noch sind keine Seiteneffekte bekannt. Es können also alle 3 Varianten verwendet werden<sup>22</sup>.

```
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
```

Listing 2.31: Standard Connector Einstellungen in tomcat/conf/server.xml

Eine mögliche Performance-Verbesserung (einstelliger Prozentbereich) ist durch die Verwendung der APR based Apache Tomcat Native library möglich, die der Tomcat-Installation beiliegt. Sobald diese in einem PATH-Verzeichnis gefunden wird, greift die Applikation darauf zurück. Ob diese Library von MediatrixWebApp-Server gefunden wird, steht in den ersten Zeilen der Logausgabe. Die Datei ist tcnative.dll und muss in einem der angegebenen Pfade liegen.

## 5.2 E-Mail-Verarbeitung (Monitoring)

E-Mails werden über den Sekundärdienst „Emaildaemon“ abgeholt und dem System zur Verfügung gestellt. Dabei ist der Workflow stets der wie in [Abbildung 13](#) dargestellt.

Aus Systembetriebssicht ist diese Verarbeitung nicht wartungsfrei, da es aufgrund der vielfältigen Formatierungen in E-Mails noch zu diversen Ausnahme-Fehlern kommen kann. Die lokale Administration sollte deshalb die folgenden Einheiten in ein Monitoring aufnehmen.

- E-Mail-Konto: Liegen E-Mails seit mehr als 10min auf dem Konto?
  - o Ja: Warnung
  - o Nein: alles OK
- Tabelle: emailscheduling: liegen dieselben Einträge hier seit länger als 1 Tag?
  - o Ja: Warnung
  - o Nein: alles OK
- Tabelle: emailscheduling\_log: übersteigt die Größe der Tabelle eine bestimmte Grenze?
  - o Ja: Warnung / Einstellungen anpassen
  - o Nein: alles OK

Aufgrund diverser Formatierungsprobleme kann es sein, dass E-Mails nicht in die Verarbeitung von Mediatrix aufgenommen werden können. Die Verarbeitung bleibt also zwischen Schritt 3 und Schritt 4 aus [Abbildung 13](#) hängen.

Damit diese Mails nicht in „emailscheduling“-Tabelle verbleiben, sollten diese in diesem Falle manuell verarbeitet werden. Allerdings kommt dieses Szenario nahezu nur bei Spam-E-Mails vor. ITyX stellt bei Bedarf Dokumentation zur Verfügung, wie diese E-Mails aus den Inhalten der Tabelle „emailscheduling“ wiederhergestellt werden können.

In der Tabelle „emailscheduling\_log“ liegen alle E-Mails, die jemals abgeholt wurden, als Kopie. Auf Dauer wird in dieser Tabelle viel Speicherplatz benötigt. Aus diesem Grund gibt es ein konfigurierbares Housekeeping in der Datei conf/mediatrix.properties, welches über diese beiden Schalter eingestellt wird

- emaildaemon.logentry.maxage (default: 90 Tage)
- emaildaemon.logentry.starttime (default: 00:00 Uhr)

Wenn Sie die E-Mail-Kopien länger oder möglichst gar nicht behalten wollen, sind diese Schalter entsprechend anzupassen

Beispiele:

```
# Lösche E-Mail-Kopien (Backups) nach 30 Tagen
emaildaemon.logentry.maxage=30
# Beginne den Löschprozess um 23:30 Uhr (täglich)
emaildaemon.logentry.starttime=23:30
```

```
# Lösche E-Mail-Kopien (Backups) nach 10 Jahren
emaildaemon.logentry.maxage= 3650
# Beginne den Löschprozess um 23:30 Uhr (täglich)
emaildaemon.logentry.starttime=23:30
```

---

*Hinweis: das hier beschriebene Housekeeping bezieht sich ausschließlich auf die SQL-Tabelle „emailscheduling\_log“, nicht auf andere Datenbank-Elemente.*

---

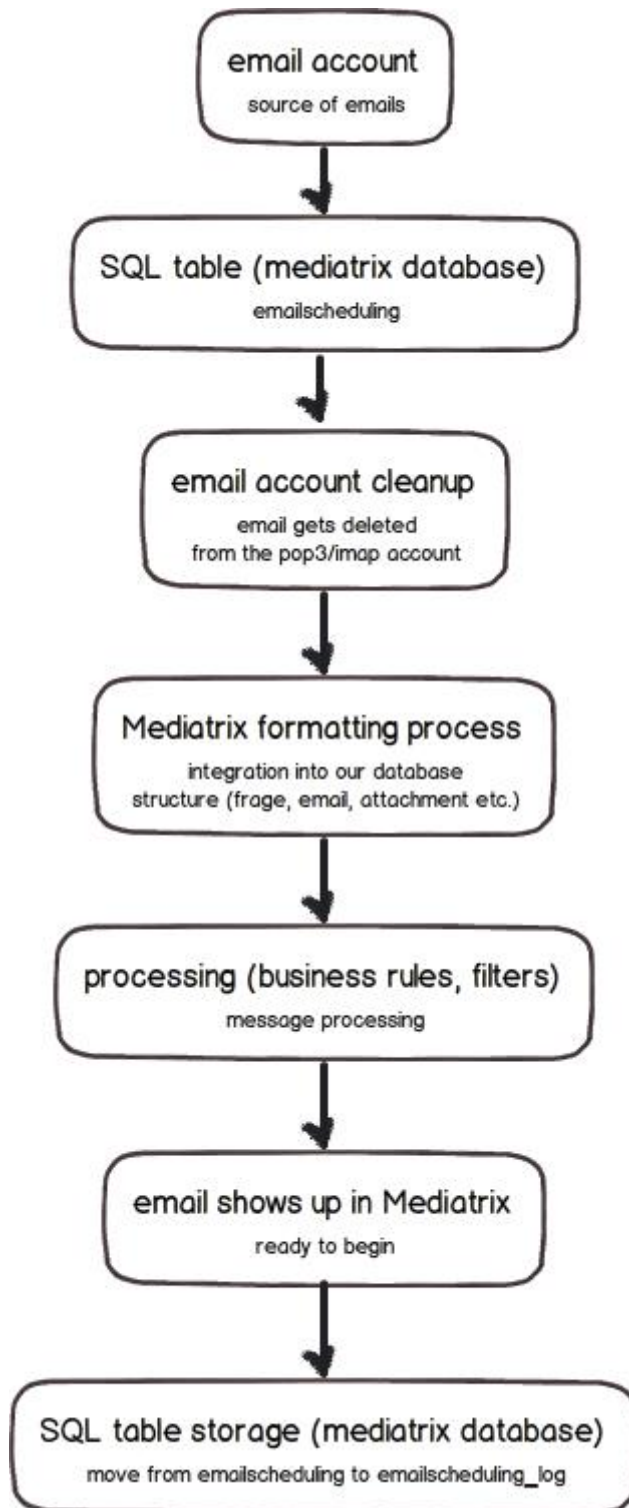


Abbildung 13: E-mail-Verarbeitung (technisch)

### 5.3 Logging

Eingesetzt wird für das Logging das Framework SLF4J mit der Implementation Log4J. Verwendet werden die Standard-Loglevel. Ein Logging auf Service-Ebene ist nicht möglich. Folgende Log-Level sind verwendbar

- ALL (Nicht empfohlen die Stufe Debug ist ausreichend)
- DEBUG
- INFO (Default – Einstellung bei einer Auslieferung)
- WARN
- ERROR
- FATAL
- OFF (Abschalten wird nicht empfohlen)

---

*Fehler-Meldungen werden in allen Log-Dateien immer über den Begriff „Exception“ angeführt. Das Vorkommen und die Bewertung von Exceptions muss über den Betrieb zumindest im Auge behalten werden. Über der Deutung von Exceptions kann der ITyX Support Auskünfte geben.*

---

#### 5.3.1 Konfiguration in Mediatix

In der mediatrix.properties (Verzeichnis: mediatrix/conf/server) existiert ein Bereich Log Configuration dort sind für das Logging folgende Schalter einstellbar:

- logpath = Angabe des Verzeichnisses für die Logs
- log.level.archive = Log-Level für Daten die in das archivdienst.log geschrieben werden
- log.level.db = Log- Level für Daten die in das sequenzer.log geschrieben werden
- log.level.email = Log- Level für Daten die in das emaildemon.log geschrieben werden
- log.level.escalation = Log- Level für Daten die in das escalation.log geschrieben werden
- log.level.outbound = Log- Level für Daten die in das outbound.log geschrieben werden
- log.level.ptme = Log- Level für Daten die in das ptme.log geschrieben werden
- log.level.webapp = Log- Level für Daten die in das servlet.log geschrieben werden
- log.filesize = Größe der Log-Dateien bevor eine neue erzeugt, bzw. die alte Datei überschrieben wird (Default 20MB).
- log.keep = Anzahl der Dateien die von dem jeweiligen log aufgehoben werden (Default 10).
- log.debugroll = Schaltet den DebugRollingFileAppender ein, ein Mediatix interner Appender (Default 1).

#### 5.3.2 Konfiguration in Contex

Um die Logausgaben zu konfigurieren gibt es eine eigene Property Datei. Im Verzeichnis „mediatrix/conf“ liegt die Datei „contex.log4j.properties“ über die die Logausgaben von Contex gesteuert werden diese kann mit Log4J Parametern konfiguriert werden.



### 5.3.3 Logdateien der einzelnen Wrapper

Neben den unter 1) und 2) benannten Log Dateien werden zusätzlich von dem jeweiligen Wrapper noch weitere Log-Dateien erzeugt. Diese beginnen jeweils mit dem Prefix „ityx“ und beinhalten neben den Ausgaben die auch in den entsprechenden Log-Dateien zu finden sind noch Ausgaben die von dem jeweiligen Wrapper erzeugt werden. Die Konfiguration erfolgt in den jeweiligen Wrapper Konfigurationsdateien (Verzeichnis: mediatrix/conf/service). Eine allgemeine Konfiguration, die für alle Dienste gültig ist, steht in der Datei „ityxcommon.inc.conf“. Diese Konfiguration kann bei Bedarf für alle Dienste in den entsprechenden „.conf“ Dateien einzeln gesetzt werden. Folgende Einstellungen sind hier möglich:

- wrapper.console.format = Format der LogAusgaben in der Konsole (Default: PM)
- wrapper.console.loglevel = Log Level für Ausgaben in der Konsole (Default: INFO)
- wrapper.logfile.format = Format der Logausgaben in der entsprechenden Datei (Default: LPTM)
- wrapper.logfile.loglevel = Log Level für Ausgaben in der entsprechenden Datei (Default: INFO)
- wrapper.logfile.maxsize = Größe der einzelnen Log Dateien (Default: 20m)
- wrapper.logfile.maxfiles = Anzahl an Log Dateien (Default: 5)
- wrapper.logfile = Verzeichnis in das die Log Dateien geschrieben werden und Angabe des Namens der Log Dateien (Default: ../../logs/%SERVICE%.log)
- wrapper.syslog.loglevel = Level für Eintragungen in das Windows Event-Log bzw. das syslog auf Unix Systemen (Default: NONE)

### 5.4 Dokumenten-Zustände (Status)

Der Verarbeitungszustand von Dokumenten, seien sie empfangen („Fragen“) oder gesendet („Antworten“), wird über den „Status“ definiert. In den [Die Kontrollleiste](#) (Kapitel 1.4.12, Seite 34), die im Mediatrix-Client abrufbar sind, können diese Zustände einzeln oder zusammengefasst angezeigt werden. Der „Status“ wird im Mediatrix-Client und in der Datenbank folgendermaßen definiert. Zur Erinnerung noch einmal Abbildung 215, die die Kontrollleiste von Mediatrix erläutert.



Abbildung 215: Mediatrix Kontrollleiste - Bedienelemente (1)

Die SQL-Queries sind für Administratoren gedacht, um bei eventuellen Analysen behilflich zu sein. Die Selektionen müssen nicht immer mit der Ansicht in Mediatrix übereinstimmen, da Mitarbeiter auch Einschränkungen besitzen könnten, die auf der Datenbank nicht gelten (Zuständigkeiten, Rechte).

Alle

Über den Status „Alle“ werden diejenigen Fragen angezeigt, die derzeit nicht im Status

- [Validierung Hold](#)
- [Validierung Extraktion](#)
- [Validierung Klassifizierung](#)
- [Externe Klassifizierung](#)
- [Erledigt](#)

befinden. Alle anderen Fragen werden über „Alle“ angezeigt, sofern die Berechtigungen und Zuständigkeiten für die Fragen gegeben sind. „Alle“ ist neben dem Status [Erledigt](#) in der Regel die größte Selektion von Fragen, weshalb in Projekten mit vielen Fragen darauf zu achten ist, ob das Mail-Inbox-Limit bereits erreicht ist. Dieses liegt standardmäßig bei 7500 Fragen, kann aber über die Kontrollleiste auf 15000 erhöht.

Über eine administrative Änderung in den Konfigurationsdateien kann das Limit nach oben und unten korrigiert werden, sowohl global als auch nur für bestimmte Mediatrix-Mitarbeiter.

„Alle“ ist ein Sammel-Zustand, es gibt keinen spezifischen Eintrag in der frage-Tabelle im Attribut status bzw. globalerstatus.

#### Wartet auf Bearbeitung

Alle Dokumente, die bearbeitet werden können und die im Operatormodus auch zugewiesen werden könnten - d.h. nur diejenigen Wiedervorlagen, die auch zeitlich „abgelaufen“ sind.

```
select * from frage
where status <>'erledigt'
and status <>'klassifikation'
and status <>'extraktion'
and status <>'hold_extraktion'
and status <>'blockiert'
and wiedervorlagezeit <CURRENT_TIME();
```

#### Bearbeitbar

Entspricht im eigentlichen Sinne wie „Alle“ Dokumente, nur dass hier zusätzlich die externen Weiterleitungen mit Rückantwort nicht mit angezeigt werden

```
select * from frage
where status <>'erledigt'
and status <>'klassifikation'
and status <>'extraktion'
```

```
and status <>'hold_extraktion'
and status <>'blockiert'
and status<>'extern-weitergeleitet';
```

## Neu

„Neu“ zeigt alle Fragen an, bei denen noch keine Bearbeitung stattgefunden hat. Das betrifft Fragen die sowohl gelesen (normale Schriftart) als auch ungelesen (fettgedruckt) dargestellt werden. Eine Bearbeitung wird dann registriert, sobald ein Mitarbeiter die Frage weiterleitet, eine Antwort speichert oder die Frage in irgendeiner Art beantwortet.

„Neu“ ist ein Zustand, der in der Datenbank (frage-Tabelle, Status-Attribut) wiedergefunden werden kann.

```
select * from frage where status='neu'or status='teilbeantwortet';
```

## Ungelesen

Zeigt nur ungelesene Fragen in der Mail-Inbox an. Ungelesene Fragen sind nie bearbeitet oder gelesen worden.

Die Nachvollziehbarkeit auf der Datenbank setzt sich folgendermaßen zusammen: Eine Selektion von Fragen

- die bisher nicht bearbeitet worden sind
- die keine Weiterleitungen sind (origin=0, ansonsten stünde hier eine Frage-ID der weitergeleiteten Original-Frage)
- die den Status „neu“ haben

```
select * from frage
where (bearbeitungszeit=0 or bearbeitungszeit is null)
and origin=0
and status='neu'
```

## Eigene Wiedervorlage

Diese Selektion zeigt nur diejenigen Fragen an, die dem aktuell angemeldeten Mitarbeiter wiedervorgelegt sind. Wiedervorlagen anderer Mitarbeiter oder nicht-wiedervorgelegte Fragen werden nicht angezeigt. Auf Datenbank-Ebene lässt sich die Anzeige wie folgt nachvollziehen. Als Mitarbeiter wurde die ID „1“ (Administrator) verwendet:

```
select * from frage where reserviertfuer=1;
```

Für andere Mitarbeiter ist entsprechend der Wert für „reserviertfuer“ zu ändern. Siehe auch [Wiedervorlage](#) weiter unten.

### Wiedervorlage

Diese Selektion zeigt nur diejenigen Fragen an, die dem allen Mitarbeitern wiedervorgelegt sind. Wiedervorlagen anderer Mitarbeiter werden angezeigt, nicht-wiedervorgelegte Fragen, oder Fragen, die dem Teilprojekt wiedervorgelegt wurden, werden nicht angezeigt.

Auf Datenbank-Ebene lässt sich die Anzeige wie folgt nachvollziehen:

```
select * from frage where reserviertfuer>0;
```

Wiedervorlagen am Teilprojekt werden nicht speziell reserviert, d.h. reserviert=0. Hier wird lediglich der Zeitpunkt gesetzt, wann eine Frage wieder zur Bearbeitung vorgelegt werden soll (wiedervorlagezeit=xxx)

```
select * from frage where reserviert=0 and wiedervorlagezeit>0
```

Wie bei jedem Status spielt auch hier die Zuständigkeit eine Rolle. Wenn eine Frage einem Mitarbeiter wiedervorgelegt ist, diese sich aber in einem Teilprojekt befindet, in dem der Mitarbeiter keine Zuständigkeit hat, so kann diese Frage nie bearbeitet werden. Die Lösung für dieses Problem ist die Wiedervorlage an das entsprechende Teilprojekt, damit andere Mitarbeiter auf diese Frage zugreifen dürfen.

Siehe auch [Eigene Wiedervorlage](#) weiter oben.

### Bearbeitet

Der Status „bearbeitet“ zeigt alle Fragen an, die weder „neu“ noch „erledigt“ sind. Für diese Fragen wurde bereits eine Bearbeitungszeit registriert, d.h. die Frage wurde (teil-)beantwortet oder es zumindest eine Antwort gespeichert.

```
select * from frage where bearbeitungszeit>0 and status<>'erledigt';
```

### Wiedervorlage (SC)

Requeue-Fall im Falle einer neuen Mail und überwachte Fragen, die vom Supervisor wiedervorgelegt wurden.

```
select * from frage where status='wiedervorlage'and servicecenter=1;
```

### Weitergeleitet

Alle Fragen, die in irgendeiner Weise weitergeleitet wurden.

```
select * from frage where weitergeleitetan<>'';
```

### Extern Weitergeleitet

Externe Weiterleitung mit Rückantwort. Diese Mail bleibt so lange in dem Status, bis die externe Antwort eintrifft.

```
select * from frage where status='extern-weitergeleitet';
```

### Extern Beantwortet

Fragen des Typs „weitergeleitet mit Rückantwort“, bei denen auch eine Rückantwort eingegangen und als Antwortfragment gespeichert wurde.

```
select * from frage where status='extern-beantwortet';
```

### Teilbeantwortet

Alle Fragen, die mindestens ein Antwort-Fragment beinhalten.

```
select distinct frage.* from frage
left join antwort on frage.id=antwort.frageid
where antwort.id is not null
and frage.orgin=0
and frage.status<>'erledigt';
```

### Erledigt

Alle erledigten Fragen. Je nach System, kann diese Abfrage länger dauern.

```
select * from frage where status='erledigt';
```

### Überwacht

Überwachte Fragen - wenn ein Mitarbeiter das Flag "überwacht" hat, dann werden von ihm bearbeitete Fragen auf „überwacht“ gesetzt. Antworten dieser Art sind im Status „ueberwacht“.

```
select * from frage join antwort on frage.id=antwort.frageid where
antwort.status='ueberwacht';
```

### Überwacht (SC)

ServiceCenter-Mails (Initialmails), die von überwachten Mitarbeitern versendet wurden.

```
select * from frage where status='ueberwacht'and frage.servicecenter=1;
```

### Validierung Hold

```
select * from frage where globalerstatus='manuelle Nachbearbeitung'and
status='hold_extraction'
```

### Validierung Extraktion

[Validierung Extraktion](#) und [Validierung Klassifizierung](#) unterscheiden sich nur in ihrer Fähigkeit bestimmte Business Rules aufzurufen. Der Funktionsumfang ist ansonsten nahezu identisch.

Fragen erhalten diesen Status, wenn über Context ein Validation-State aufgerufen wird, der den Context-Prozess unterbricht. In der Konfiguration des Validation-States muss „Extraction“ eingestellt werden, damit ein Context-Dokument in Mediatrix im Status [Validierung Extraktion](#) erscheint. In der Regel wird dieser Status verwendet, um Tagmatches, die in Context nicht automatisch ermittelt werden konnten, durch einen Agenten manuell anzureichern. Nachdem das Dokument mit zusätzlichen Daten gespeichert wurde, wird es in Mediatrix auf den Status „erledigt“ gesetzt und Context wird den Prozess mit den neuen Informationen weiterverarbeiten.

Hinweis: Im Falle, dass Context als „e-Kat“ (Kapitel [1.5.10](#), Seite [90](#)) eingesetzt wird, ist zu beachten, dass Mediatrix hier zwei getrennte Frage-Objekte erhalten wird; eine Frage im Status [Externe Klassifizierung](#) und eine im Status [Validierung Extraktion](#). Diese Objekte werden unterschiedliche Frage-IDs haben!

Auf Datenbankebene findet man Fragen dieses Status' mit folgender Query:

```
select * from frage where globalerstatus='manuelle Nachbearbeitung'and
status='extraktion'
```

### Validierung Klassifizierung

[Validierung Extraktion](#) und [Validierung Klassifizierung](#) unterscheiden sich nur in ihrer Fähigkeit bestimmte Business Rules aufzurufen. Der Funktionsumfang ist ansonsten nahezu identisch. Fragen erhalten diesen Status, wenn über Context ein Validation-State aufgerufen wird, der den Context-Prozess unterbricht. In der Konfiguration des Validation-States muss „Classification“ eingestellt werden,

damit ein Context-Dokument in Mediatrix im Status [Validierung Klassifizierung](#) erscheint. In der Regel wird dieser Status verwendet, um die Klassifizierung des Dokuments, die in Context nicht automatisch ermittelt werden konnte, durch einen Agenten manuell anzureichern. Nachdem das Dokument mit zusätzlichen Daten gespeichert wurde, wird es in Mediatrix auf den Status „erledigt“ gesetzt und Context wird den Prozess mit den neuen Informationen weiterverarbeiten.

Hinweis: Im Falle, dass Context als „e-Kat“ (Kapitel [1.5.10](#), Seite [90](#)) eingesetzt wird, ist zu beachten, dass Mediatrix hier zwei getrennte Frage-Objekte erhalten wird; eine Frage im Status [Externe Klassifizierung](#) und eine im Status [Validierung Klassifizierung](#). Diese Objekte werden unterschiedliche Frage-IDs haben!

Auf Datenbankebene findet man Fragen dieses Status' mit folgender Query:

```
select * from frage where globalerstatus='manuelle Nachbearbeitung' and
status='klassifikation'
```

### Externe Klassifizierung

Wenn für ein Mediatrix-Projekt die e-Kat-Funktion (Kapitel 1.5.10, Seite 90) eingesetzt wird, so werden Mediatrix-Fragen zunächst in den Zustand 'Externe Klassifizierung' versetzt und somit in der Ansicht „Alle“ ausgeblendet. Sobald die externe Klassifizierung abgeschlossen ist, werden diese Fragen über den Emailedemon-Dienst während des regulären Abholintervalls aus der externen Klassifizierung abgeholt und an die Mediatrix-Verarbeitung übergeben.

```
select * from frage where status='blockiert';
```

Sollten Fragen nicht aus der Externen Klassifizierung zurück kommen, weil es hier zu einem Fehler gekommen ist, so können die Fragen über den Status [Externe Klassifizierung](#) manuell an die reguläre Mediatrix-Verarbeitung übergeben werden. Dabei wird die Klassifizierung übersprungen, aber das Dokument kann bearbeitet werden.

### Gelöscht

Es handelt sich hier um einen Sonder-Status. Gelöschte Fragen können weder im Client angezeigt werden, noch spielen diese in Reports noch eine Rolle. Fragen können aus der Inbox über Rechtsklick → Löschen entfernt werden.

Gelöschte Fragen können im Client nicht direkt angezeigt oder wiederhergestellt werden, für eine Wiederherstellung ist ein Zugriff auf die Datenbank notwendig.

Fragen werden über diese Aktion lediglich logisch gelöscht, sie sind über einen Datenbank-Administrator wiederherstellbar.

Query: `select * from frage where geloeschtam > 0;`

## 5.5 Performance Monitoring

Dieses Programm kann im Mediatrix-Client gestartet werden. Es hat zwei Komponenten

- Hintergrund-Messung
- Monitor-GUI

Startet man die GUI, den „Performance Monitor“, so wird die Hintergrund-Messung automatisch mitgestartet. Standardmäßig wird die Messung nicht beim Start des Clients aktiviert, da die Messung ebenfalls Leistung benötigt, die man nur dann opfern sollte, wenn es tatsächlich etwas zu analysieren gibt.

Während der Hintergrundmessung wird im Benutzer-Verzeichnis des PCs, auf dem Mediatrix-Client gestartet ist, eine Datei geschrieben, die die erhobenen Daten im CSV-Format speichert, z.B.

```
c:\User\<username>\Mediatrix\trace.log
```

Diese Datei entspricht im Aufbau der Standard-Darstellung des Performance-Monitors.

Date	Function	Client->Server (kB)	Time/Client (ms)	Server->Client (kB)	Time/Server (ms)	Deadlock-Prevention
15:50:58.896	echo	0.249kB	62ms	8.455kB	4ms	0
15:50:58.928	echo	0.249kB	32ms	8.453kB	4ms	0
15:50:58.975	echo	0.249kB	47ms	8.456kB	4ms	0
15:50:59.021	echo	0.249kB	46ms	8.454kB	3ms	0
15:50:59.100	echo	16.407kB	63ms	0.249kB	0ms	0
15:50:59.193	echo	16.409kB	78ms	0.249kB	0ms	0
15:50:59.271	echo	16.408kB	62ms	0.249kB	0ms	0
15:50:59.350	echo	16.405kB	63ms	0.249kB	0ms	0
15:50:59.412	echo	0.249kB	62ms	16.408kB	6ms	0
15:50:59.475	echo	0.249kB	63ms	16.407kB	6ms	0
15:50:59.553	echo	0.249kB	78ms	16.407kB	7ms	0
15:50:59.615	echo	0.249kB	62ms	16.406kB	7ms	0
15:50:59.771	echo	32.299kB	125ms	0.249kB	0ms	0
15:50:59.947	echo	32.298kB	145ms	0.249kB	0ms	0
15:51:00.103	echo	0.249kB	150ms	32.295kB	13ms	0
15:51:00.228	echo	0.249kB	125ms	32.294kB	12ms	0
15:51:26.408	ping	0.005kB	31ms	0.134kB	9ms	0
15:51:26.424	getNotifyMessage	0.237kB	16ms	0.005kB	1ms	0
15:51:26.424	multipleCalls	0.573kB	16ms	0.252kB	4ms	0
15:52:35.657	getResultDirectly	0.438kB	16ms	2.218kB	10ms	0
15:54:26.426	ping	0.005kB	16ms	0.134kB	4ms	0
15:54:26.457	getNotifyMessage	0.237kB	31ms	0.005kB	1ms	0
15:54:26.457	multipleCalls	0.573kB	31ms	0.252kB	2ms	0
15:57:26.443	ping	0.005kB	16ms	0.134kB	4ms	0
15:57:26.458	getNotifyMessage	0.237kB	15ms	0.005kB	1ms	0
15:57:26.458	multipleCalls	0.573kB	15ms	0.252kB	2ms	0
15:57:50.901	getResultDirectly	0.438kB	31ms	2.218kB	13ms	0
16:00:26.477	ping	0.005kB	31ms	0.134kB	7ms	0
16:00:26.509	getNotifyMessage	0.438kB	32ms	0.005kB	1ms	0
16:00:26.509	multipleCalls	0.573kB	32ms	0.252kB	3ms	0
16:03:06.133	getResultDirectly	0.438kB	63ms	2.218kB	20ms	0
16:03:26.489	ping	0.005kB	0ms	0.134kB	6ms	0
16:03:26.505	getNotifyMessage	0.237kB	0ms	0.005kB	1ms	0
16:03:26.500	multipleCalls	0.573kB	15ms	0.252kB	2ms	0
16:06:26.501	ping	0.005kB	0ms	0.134kB	3ms	0
16:06:26.517	getNotifyMessage	0.237kB	16ms	0.005kB	1ms	0
16:06:26.517	multipleCalls	0.573kB	16ms	0.252kB	3ms	0

Abbildung 14: Mediatrix-Performance-Monitor GUI

In diesem Fenster wird jede Kommunikation zwischen Client und Server aufgezeichnet. Die Zeit und das Datenaufkommen werden dabei festgehalten.

Spalte	Bedeutung
<b>Date</b>	Zeitpunkt der Aktion
<b>Function</b>	interner Funktionsname; z.B. das Laden der Mailinbox-Liste heist "getResult"
<b>Client-&gt;Server (kB)</b>	Datenübertragung vom Client zum Server in Kilobytes.  <u>Hinweis:</u> in der Datei trace.log sind Bytes angegeben
<b>Time/Client (ms)</b>	Gesamtzeit für die Ausführung der Funktion (vom Kommando des Users bis zur Ausführung)
<b>Server-&gt;Client (kB)</b>	Datenübertragung vom Server zum Client in Kilobytes.  <u>Hinweis:</u> in der Datei trace.log sind Bytes angegeben
<b>Time/Server (ms)</b>	Zeit, die serverseitig benötigt wurde, um ein Kommando zu bearbeiten (z.B. „suche erledigte E-Mails mit dem Betreff „Rechnung““)



<b>Deadlock-Prevention</b>	Ein True/False-Wert, ob während dieses Kommandos ein Deadlock aufgetreten ist, der vom System unterbunden wurde
----------------------------	---

Die Deutung der Werte ist vielseitig und kann in der Regel von jedem Administrator durchgeführt werden. Wir führen nur die Top 3 der Performance-Indikatoren auf:

1. Wenn die Aktion auf dem Client sehr lange braucht, auf dem Server aber nur niedrige Zeit-Werte in der Spalte „Time/Server“ stehen, kann man davon ausgehen, dass es zwischen Client und Server entweder eine zu geringe Bandbreite gibt, ein Netzwerk-Problem besteht oder ein überdurchschnittlich großes Dokument übertragen werden musste. In diesem Falle sollten Time/Server und Time/Client eine Hohe Differenz haben
2. Wenn die Aktion auf dem Client sehr lange braucht, auf dem Server ebenfalls eine sehr hohe Zeit verzeichnet ist, dann ist davon auszugehen, dass entweder der Applikations-Server ein Performance-Problem hat (RAM,CPU,Harddisk-I/O ausgelastet), die Berechnungslogik ungünstig ist, in der Verarbeitung des Kommandos ein langsames Drittsystem beteiligt ist, die Datenbank keine Indizes verwendet oder ein Problem hat oder ein Software-Problem vorliegt
3. Wenn das Öffnen von Dokumenten schnell geht, das Versenden allerdings lange dauert, dann kann das daran liegen, dass entweder zwischen Client und Server eine Asynchrone Netzleitung anliegt (z.B. DSL, 16Mbit download, 1Mbit upload) oder der Server keine schnelle Verbindung zum Postausgangs-Server aufbauen kann (E-Mails werden durch den Server in Echtzeit versendet, während der Server sendet kann es zu Wartezeiten bei demjenigen Client kommen, der den Sendeauftrag erteilt hat)

Neben der tabellarischen Darstellung von Performance-Indikatoren kann Mediatrix über den Tab „System Load Chart“ eine Kurz-Analyse durchführen, welche Übertragungsrate bei welcher Paketgröße möglich ist. Diese Überprüfung kann mit einem Klick auf den Button „System Load Test“ erzeugt werden und dauert in der Regel nur wenige Sekunden.

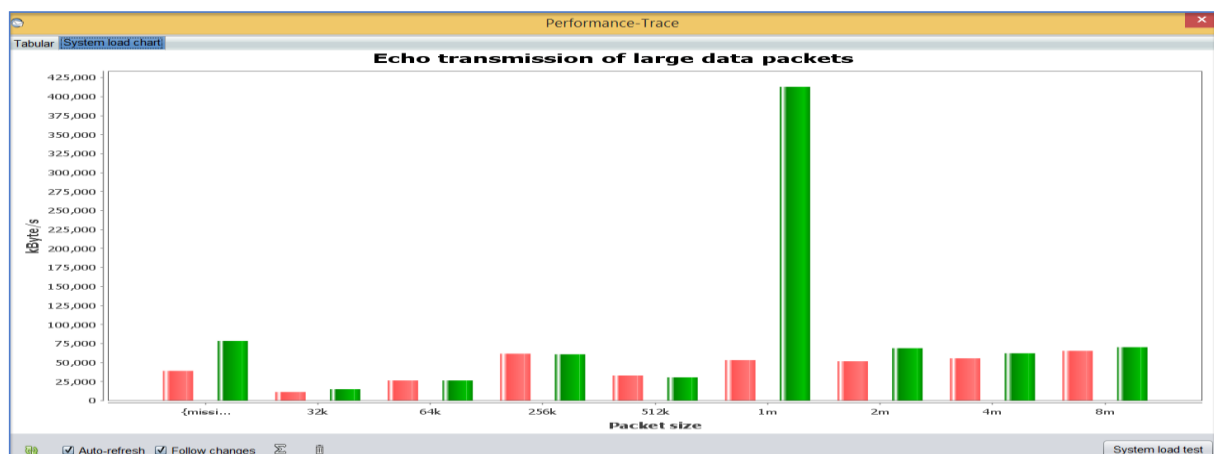


Abbildung 15: Mediatrix-Performance-Monitor Netzwerk-Messung

## mediatrix.properties

Voraussetzung für die Nutzung dieser Funktion sind Einstellungen in der Datei `mediatrix.properties`:

```
trace.size=1
trace.timing=1
```

### trace.size

`trace.size=1` (default 0)

Aktiviert die Funktionalität des Performance-Monitors im Mediatrix-Client. Diese Funktion sollte nur von einem Client im Netzwerk gleichzeitig verwendet werden, weil die Messung auch Leistung benötigt. Der Performance-Monitor wird über den Client im Menüpunkt → Extras → Performancemessung aktiviert. `trace.size` sollte immer in Verbindung mit `trace.timing` verwendet werden.

### trace.timing

`trace.timing=1` (default 0)

Aktiviert die Funktionalität des Performance-Monitors im Mediatrix-Client. Diese Funktion sollte nur von einem Client im Netzwerk gleichzeitig verwendet werden, weil die Messung auch Leistung benötigt. Der Performance-Monitor wird über den Client im Menüpunkt Hilfe → Extras → Performancemessung aktiviert. `trace.timing` sollte immer in Verbindung mit `trace.size` verwendet werden.

## client.properties

Ist das Verhalten „alle Mediatrix-Clients sollen ab Start Messungen durchführen“ erwünscht, kann über einen Eintrag in der Server-Konfiguration `conf/custom/client.properties` die Messung für Clients schon ab dem Start forciert werden

### de.ityx.mediatrix.PerformanceTrace

`de.ityx.PerformanceTrace=true` (default false)

Aktiviert direkt beim Start eines Clients die Performanz-Messung, sobald sich ein Agent eingeloggt hat. Diese Einstellung sollte nur temporär global eingesetzt werden, da jeder Client, der eine Leistungsmessung durchführt, auch jeweils etwas Leistung kostet. Die Ergebnisse der Messung werden im Mediatrix-Verzeichnis auf der Client-Seite (pro Workstation im User-Directory) gespeichert (`trace.log`)