

1 Manuelle Konfiguration

Diese Einführung ist für Fachleute gedacht, die in der Lage sind, die manuelle Konfiguration für das MEDLOG-System professionell durchzuführen.

Als erstes werden die Konfigurationsdateien vorgestellt und erläutert, es folgen die Lizenzdateien sowie eine Beschreibung für die Durchführung einer Demo-Installation. Zum Schluss folgen Hinweise für eine mögliche alternative Konfiguration.

Dieses Kapitel basiert auf den Kapiteln 4 (Installation) und 5 (Geräte anschließen und konfigurieren) des Benutzerhandbuchs.

Wichtiger Hinweis: Bevor mit der Konfiguration begonnen wird, sind die Warnhinweise bezüglich Verkabelung, Installation, Bedienung und Integration des MEDLOG-Systems in Kapitel 5 sorgfältig durchzulesen!

1.1 Konfigurationsdateien

Die Konfigurationsdateien des Servers sind im Installations-Verzeichnis „conf/server“ gespeichert. Der vollständige Pfad bei der Installation ist standardmäßig:
C:\Programme\BMC\conf\server

Die Konfigurationsdateien sind in der Syntax XML (Extensible Markup Language) gehalten. XML ist so aufgebaut, dass sämtliche Einstellungswerte in so genannten Tags enthalten sind. Tags werden mit spitzen Klammern notiert, z.B. <connection>. Auf ein öffnendes Tag folgt ein schließendes Tag, welches zusätzlich ein '/' vor dem bezeichnenden Namen enthält, z.B. </connection>. Ein komplettes XML-Element wird dann folgendermaßen notiert:

<connection>hier Einstellungen</connection>.

Tags können entweder Werte enthalten oder wiederum Tags. Dadurch können Tags gruppiert werden:

```
<connection>  
    <databaseName>medlog</databaseName>  
</connection>
```



Hinweis: Sollen Einstellungen an den Konfigurationsdateien verändert werden, dann können NUR die Daten, Werte und Einstellungen zwischen den Tags verändert werden. Es ist **nicht** erlaubt, die Tag-Struktur umzustellen, es dürfen lediglich die Werte zwischen den Tags angepasst werden.

Soll beispielsweise der Name für die Datenbank nicht „medlog“ lauten, sondern „healthcare“, dann müsste das oben angegebene Beispiel folgendermaßen verändert werden:

```
<connection>  
    <databaseName>healthcare</databaseName>  
</connection>
```



Hinweis: Keinesfalls darf ein Tag geöffnet und nicht wieder geschlossen werden oder umgekehrt. Ebenso führen Klammerfehler zu einer ungültigen Konfiguration. Bei Unsicherheiten im Umgang mit XML-Konfigurationsdateien wird empfohlen, den Administrations-Editor zu verwenden.

In der folgenden Tabelle werden alle Konfigurationsdateien mit ihrer jeweiligen Bedeutung aufgelistet. Mit einem Sternchen „*“ markierte Konfigurationsdateien müssen vor dem ersten Betrieb von MEDLOG angepasst werden. Die übrigen Konfigurationsdateien können bei Bedarf geändert werden.



Hinweis: Vor dem Ändern sollten Kopien der Konfigurationsdateien erstellt werden und in einem sicheren Speichermedium untergebracht werden, damit bei fehlerhaften Änderungen der alte Zustand jederzeit wieder hergestellt werden kann. Dies ist insbesondere wichtig, falls Änderungen einen Verlust der Lizenz mit sich bringen.

Konfigurationsdatei	Bedeutung
database.xml*	Einstellungen für die Datenbank
device.xml*	Einstellungen für die angeschlossenen Geräte
extern-interface.xml	Verwendung von Schnittstellen für den Zugriff auf gespeicherte Parameter
filter.xml	Filtereinstellungen und Einheitenumwandlungen
log.xml	Einstellungen für Logging-Mechanismus (es wird log4j verwendet)
servlet.xml	Einstellungen für die HTTP-basierte XML-Schnittstelle
sslserver.xml	Zugriff vom Administrations-Editor
bedidassignment.xml	Zuweisungen von Bett-IDs auf USB-Ports. Diese Datei kann nicht von Hand erstellt werden, sondern muss über den Administrations-Editor erzeugt und editiert werden (der Editor fragt USB-Kennungen des Servers ab.)
serverlist.xml	Liste aller weiteren MEDLOG-Installationen, an die Anfragen weitergereicht werden sollen

1.1.1 database.xml

Über die Konfigurationsdatei „database.xml“ können sämtliche datenbankspezifischen Eigenschaften eingestellt werden. Eine vollständige Fassung dieser Konfiguration mit allen wichtigen Einstellungen sieht beim Öffnen dieser Datei folgendermaßen aus:

```
<database disabled="false">
  <connection>
    <databaseName>medlog</databaseName>
    <url>jdbc:mysql://localhost:3306/</url>
    <userName>root</userName>
    <password plain="true"></password>
  </connection>
  <monitor>>false</monitor>
  <recording>
    <autostart>>true</autostart>
    <timeDifference>10</timeDifference>
    <advancedFiltering>>false</advancedFiltering>
    <maxTableSize>800000</maxTableSize>
  </recording>
  <deletion>
    <valueDateTime>-1</valueDateTime>
  </deletion>
</database>
```

Die einzelnen Tags haben folgende Bedeutungen:

• <database disabled="false">Enthält (geschachtelte) Tags</database>

Das Attribut "disabled" in diesem Tag gibt an, ob die Datenbank, in der die von medizinischen Geräten erhaltenen Daten gespeichert werden können, aktiviert ist oder nicht. Um die Datenbank zu aktivieren bzw. zu deaktivieren, muss der Wert true bzw. false eingetragen werden. Standardwert ist false, d.h., die Datenbank ist nicht aktiviert.

• <connection>Enthält (geschachtelte) Tags</connection>

Für den Zugriff auf eine Datenbank müssen die Einträge im Tag <connection> vollständig und korrekt sein. Dafür müssen folgende Parameter angegeben werden:

- Name der Datenbank: wird als String im Tag <databaseName> eingetragen
- URL für den Zugriff auf die Datenbank: wird als String im Tag <url> eingetragen (z.B. für MySQL: jdbc:mysql://<Rechnername/IP>:3306/)
- Benutzername: wird als String im Tag <UserName> eingetragen
- Passwort: wird im Tag <Passwort> eingetragen (optional kann das Passwort zusätzlich mit Base64 verschlüsselt angegeben werden; ist das Passwort zusätzlich verschlüsselt, dann muss das Attribut „plain“ auf „false“ gestellt sein)

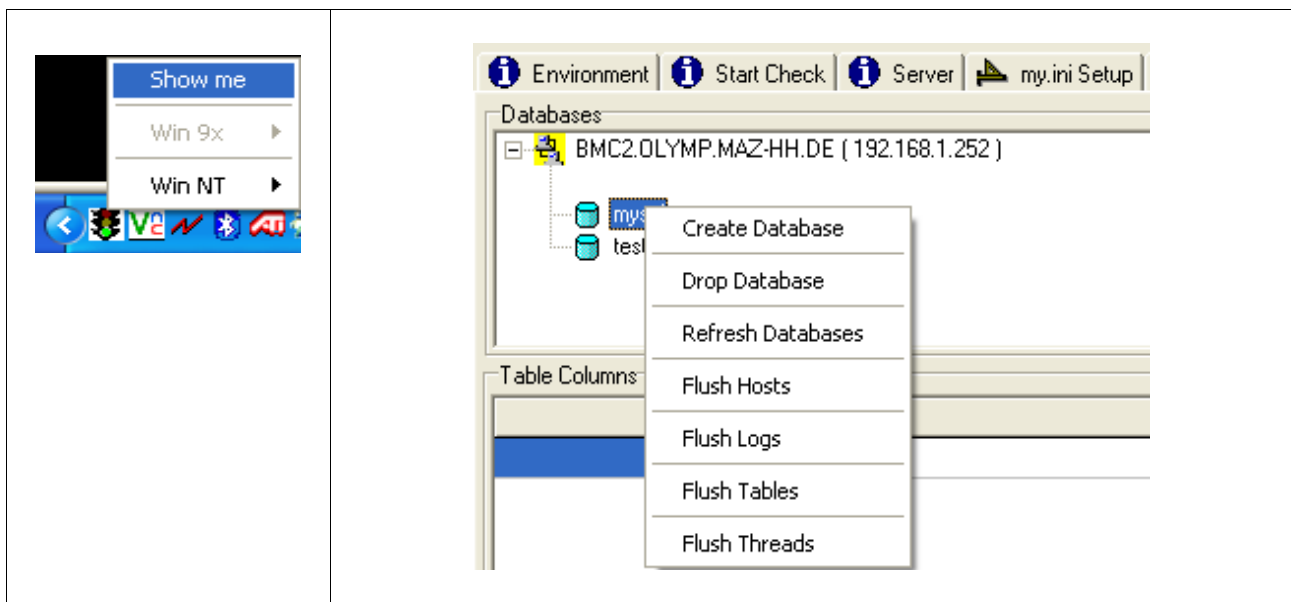
• <databaseName> Hier steht der Name der Zieldatenbank </databaseName>

Der Name einer Datenbank wird mit diesem Tag festgelegt.

Falls das angegebene Konto (Benutzername und Passwort in Tags <userName> und <Passwort>) für MySQL mit genügend Benutzerrechten ausgestattet ist und die Datenbank in MySQL noch nicht angelegt wurde, dann übernimmt der Server automatisch die Installation und legt die Datenbank selbst an.

Falls der Account jedoch nicht über diese Rechte verfügt, dann muss vor dem Start des MEDLOG-Servers die Datenbank manuell angelegt werden. Datenbanken können entweder mit dem Tool winmysqladmin.exe oder über das MySQL Control Center erzeugt werden. Bei der Installation von MYSQL ist das Tool "winmysqladmin.exe" automatisch mitgeliefert und befindet sich im "Bin"-Verzeichnis von MYSQL.

Um die Datenbank mit winmysqladmin.exe anzulegen, müssen MySQL und winmysqladmin.exe gestartet sein. Das Fenster von winmysqladmin wird nach rechtem Mausklick auf das Tray-Icon mit dem Eintrag „Show me“ angezeigt.



Dort muss der Reiter „databases“ angewählt werden. Links oben im Fenster wird ein Ausschnitt mit der Bezeichnung „databases“ angezeigt, dieser enthält einen Wurzeleintrag mit der Bezeichnung des Rechners und mindestens einen Untereintrag, der „mysql“ heißt.

Mit einem weiteren Rechtsklick auf den Wurzeleintrag wird ein Kontextmenü aufgerufen, in dem der Eintrag „Create Database“ gewählt werden sollte. Darin ist als Datenbankname derselbe Namen einzutragen, der auch im Tag <databaseName> in der Konfigurationsdatei enthalten ist (der Standardeintrag ist „medlog“).

• **<url>Hostname oder IP des Rechners, auf dem die Datenbank läuft</url>**

Das Tag <url> ist für den Zugriff auf MySQL notwendig und bezeichnet den Ort, an dem die Datenbank installiert wurde. Die Ortsangabe ist abhängig vom Namen oder der IP des Computers. Der Port, auf dem auch die Datenbank zugänglich ist, muss auch angegeben werden. Im Beispiel unten ist dies der Port 3306.

Befinden sich MySQL und MEDLOG auf demselben Rechner, dann kann wie im Beispiel „localhost“ eingetragen werden „<url>jdbc:mysql://localhost:3306/</url>“.

Ist MySQL auf einem räumlich entfernten Rechner installiert und durch ein Netzwerk verbunden, so muss statt "localhost" beispielsweise die IP dieses Rechners eingetragen werden, z.B.: `<url>jdbc:mysql://192.128.18.3:3306/</url>`.

Es sollte zuvor sichergestellt werden, dass die verbundenen Rechner untereinander kommunizieren können. Eine Firewall oder falsche Netzwerkeinstellungen könnten unter Umständen eine korrekte Verbindung behindern.

Die Portnummer und die IP-Adresse von dem Rechner, auf dem die Datenbank läuft, werden vom zuständigen Netzwerkadministrator im Krankenhaus oder im Einsatzgebiet bekannt gemacht.

• **`<userName>`Benutzername der Datenbank`</userName>`**

Hier wird der Benutzername der Datenbank eingetragen. Dieser Name wird vorher angelegt. Meist werden neue Accounts (Benutzernamen, Passwort) von dem Netzwerkadministrator bzw. Datenbankadministrator mit Hilfe der Software MySQL Control Center angelegt.

• **`<password plain="true">`Passworteingabe`</password>`**

Das Passwort des Datenbankkontos wird im Eingabefeld `<password>` gespeichert.

Wird das Passwort in Klartext angegeben, so muss der Eintrag `plain="true"` innerhalb des Tags `<password>` verwendet werden.

Falls das Passwort nicht in Klartext in der Konfiguration gespeichert werden soll, dann muss anstatt „true“ der Eintrag „false“ gewählt und das Passwort als Base64-kodierte Zeichenkette angegeben werden.

Der Administrations-Editor codiert das Passwort automatisch als Base64-Zeichenkette. Der Vorteil von Base64-codierten Passwörtern ist, dass diese nicht in Klartext angelegt und somit nicht versehentlich lesbar sind.

• **Zusätzliches Tag: `<parameters>` `</parameters>`**

Der Zugriff auf die MySQL-Datenbank kann für den Fall, dass die Datenbank und der MEDLOG-Server auf demselben Rechner installiert wurden, beschleunigt werden. Um diese Beschleunigung einzurichten, muss zusätzlich das Tag `<parameters>` mit der Einstellung aus dem unten angegebenen Beispiel hinzugefügt werden. Gleichzeitig muss bei der Angabe der URL im Tag `<url>` der Eintrag „localhost“ enthalten sein.

```

<connection>
  <databaseName>medlog</databaseName>
  <url>jdbc:mysql://localhost:3306/</url>
  <parameters>
    <parameter key="socketFactory" value="com.mysql.jdbc.NamedPipeSocketFactory"/>
  </parameters>
  <userName>root</userName>
  <password plain="true">passwort</password>
</connection>

```

Durch diese Einstellung wird die Kommunikation nicht mehr über das Netzwerk, sondern direkt über den Speicher durchgeführt.

Beim Einsatz von Netzwerken mit DHCP (dynamic host configuration protocol) muss darauf

geachtet werden, dass das MySQL und das MEDLOG-System, sowie auch das Dokumentationssystem, stets die gleichen Rechnernamen bzw. IPs zugewiesen bekommen - abhängig davon, was in der Konfiguration angegeben wurde. Nachträgliche Änderungen können zu einer ungültigen Konfiguration führen.

Weiterhin ist zu beachten, dass Rechner unter Windows sich selbst nicht mehr über ihren eigenen Rechnernamen erreichen können, sollten sie einmal vom Netzwerk getrennt werden.

Sind MySQL und MEDLOG auf einem Rechner installiert, so sollte der MySQL-Account die Freigaberechte über "localhost" bekommen und nicht über den Rechnernamen.

• **<monitor>"true" oder "false"</monitor>**

Um direkt beim Start des MEDLOG-Servers als Applikation auf die eingehenden und gespeicherten Daten zugreifen zu können, muss zuvor der Eintrag für das Tag <monitor> auf den Wert „true“ gestellt werden: <monitor>true</monitor>

Bei eingeschalteter Datenbank wird ein Fenster angezeigt, in dem die abgespeicherten Daten durchlaufend angezeigt werden. Für den Zwischenspeicher wird ein zweites Fenster geöffnet, welches die aktuellen Daten aus diesem anzeigt.

Diese Option - mit der Einstellung des MEDLOG-Servers als Applikation - ist nur für den Testbetrieb vorgesehen und sollte für eine Langzeitinstallation nicht eingeschaltet sein.

Ist der Server als Dienst installiert, werden keine Fenster angezeigt. Die Monitoranzeige sollte dennoch deaktiviert sein, da die Fenster lediglich vom Betriebssystem ausgeblendet werden. Ist es beispielsweise für einen Test erforderlich, die Fensterinhalte trotzdem darzustellen, dann muss zum einen diese Option eingeschaltet sein und zum anderen der Dienst in Windows umgestellt werden.

Diese Umstellung kann in der Übersicht der Dienste durchgeführt werden (Start -> Systemsteuerung -> Verwaltung -> Dienst). Der MEDLOG-Server muss dafür bereits als Dienst installiert sein.

In der Übersicht kann per Doppelklick ein Eigenschaftsfenster zum Dienst MEDLOG geöffnet werden, in dessen Reiter „Anmelden“ die Option „Datenaustausch zwischen Dienst und Desktop zulassen“ zur Verfügung steht. Nach Aktivierung dieser Option werden ab dem nächsten Start des Dienstes die Fenster des MEDLOG-Servers angezeigt.

Zu beachten: diese Einstellung ist nach einer Deinstallation des Dienstes wieder verloren.

• **<recording>Enthält weitere geschachtelte Tags</recording>**

Über das Tag <recording> können das Aufzeichnungsverhalten, die Filtermechanismen von MEDLOG sowie die Tabellengöße der zugrunde liegenden Datenbank gesteuert werden. Dafür sind die Tags <autostart>, <timeDifference>, <advancedFiltering> und <maxTableSize> zuständig, wie das Beispiel unten zeigt:

```
<recording>  
  <autostart>true</autostart>  
  <timeDifference>10</timeDifference>  
  <advancedFiltering>>false<advancedFiltering>  
  <maxTableSize>800000</maxTableSize>  
</recording>
```

• **<autostart>** "true" oder "false" **</autostart>**

Mit Hilfe der Einstellung <autostart> kann eingerichtet werden, dass sofort nach dem Start der MEDLOG-Serversoftware Daten in der Datenbank aufgezeichnet werden sollen. Es wird empfohlen, hier „true“ zu wählen und damit Autostart einzuschalten. Mit der Einstellung „false“ wird die Funktion deaktiviert, der Start muss dann manuell erfolgen, z.B. über die HTTP-Schnittstelle durch das Dokumentationssystem.

• **<timeDifference>** Zahlangabe zwischen 10 und 60 **</timeDifference>**

Die Einstellung <timeDifference> gibt vor, mit welcher maximalen Genauigkeit Werte in der Datenbank gespeichert werden sollen.

Der Wert wird in Sekunden angegeben. Empfängt beispielsweise ein Treiber sekundlich bestimmte Werte, so wird bei einer <timeDifference>-Einstellung von „10“ nur jeder zehnte empfangene Wert tatsächlich abgespeichert. Häufig ist es gar nicht notwendig, Werte mit einer genaueren Auflösung zu speichern. Je höher der eingestellte Wert, desto mehr wird die Datenbank entlastet, da weniger Daten abgespeichert werden.

Wird der <timeDifference>-Wert niedriger eingestellt, dann werden entsprechend mehr Parameter in der Datenbank gespeichert, allerdings wird dadurch die Datenbank stärker belastet.

Es wird empfohlen, einen Wert zwischen 10 und 60 Sekunden zu wählen.

• **<advancedFiltering>** "true" oder "false" **</advancedFiltering>**

Mit der Einstellung <advancedFiltering>true<advancedFiltering> kann ein zusätzlicher Filtermechanismus eingeschaltet werden, durch den es möglich ist, für jeden einzelnen Parameter die Samplerate anzugeben. Die Festlegung dieser Daten geschieht in der Konfigurationsdatei filter.xml.

• **<maxTableSize>** Zahleneingabe: max. zu speichernde Parameter **</maxTableSize>**

Die Datenmenge, die ständig durch die angeschlossenen Geräte erzeugt wird, ist verhältnismäßig hoch, und es kommt bei Datenbankabfragen schnell zu Performanz-Einbußen. Um dies zu verhindern, verfügt MEDLOG über einen Mechanismus, der Parameter automatisch archiviert.

Archivierte Daten können über die externen Anfrageschnittstellen weiterhin über denselben Mechanismus abgefragt werden. Da dies jedoch in der Regel seltener geschieht und Anfragen sich zumeist auf zeitnahe Bereich beziehen, werden zeitnahe Daten bevorzugt behandelt.

Mit dem Tag <maxTableSize> kann eingestellt werden, wie viele Parameter maximal in einer Tabelle gespeichert werden sollen, bevor sie archiviert werden.

• **<deletion>** Enthält ein einziges Tag **</deletion>**

Neben dem selektiven Löschen der Parameter über die externe Schnittstelle besteht die Möglichkeit, Parameter automatisch vom System löschen zu lassen, wie das Beispiel zeigt:

```
<deletion>
    <valueDateTime>-1</valueDateTime>
</deletion>
```

• **<valueDateTime>** Zahleneingabe: wann die Daten gelöscht werden **</valueDateTime>**

Mit diesem Tag wird festgelegt, wann die gespeicherten Daten gelöscht werden sollen.

Mit dem Eintrag „-1“ wird dieser Mechanismus abgeschaltet. Ist jedoch die Null oder eine positive Zahl dort eingetragen, z.B. 6, so werden alle Daten, die älter als 6 Tage sind, automatisch abends gelöscht.

• **<levels>** Enthält weitere geschachtelte Tags **</levels>**

Mit diesem Tag wird das Filter- und Verarbeitungsverhalten für Parameterdaten bestimmt.

```
<levels>
  <level name="plain">
    <handler className="medlog.db.store.LatestValuesCacheHandler"/>
  </level>
  <level name="minTimeFilter" filterClassName="medlog.db.filter.MinTimeFilter">
  </level>
  <level name="simpleParameterFilter" filterClassName="medlog.db.filter.SimpleFilter">
    <handler className="medlog.db.store.LogHandler"/>
  </level>
  <level name="advancedFilter" filterClassName="medlog.db.filter.AdvancedFilter">
    <handler className="medlog.db.store.MySqlHandler"/>
  </level>
</levels>
```

• **<level>** Name plus optional dem Namen der Filterklasse **</level>**

Dies Tag definiert den Filtertyp. Die folgenden Ebenen sind verfügbar:

Tag-Wert	Bedeutung
plain:	keine Filterung
minTimeFilter:	Filterung nach geringstem Alter von Werten
simpleParameterFilter:	Filterung nach Parameterliste (filter.xml)
advancedFilter:	Erweiterte Filterung anhand der Differenz zum letzten gespeicherten Wert

• **<handler>Name</handler>**

So genannte 'StoreHandler' können auf den unterschiedlichen Ebenen definiert werden; sie leiten gefilterte Daten an die entsprechenden Ebenen weiter. Ein Handler wird in die Ebene eingehängt, indem das Tag `<handlerclassName="used Handler class"/>` hinzugefügt wird.

Mögliche Handler sind:

- LatestValuesCacheHandler: medlog.db.store.LatestValuesCacheHandler (= name)
zur Datenspeicherung im Online-Zwischenspeicher (Cache)
- medlog.db.store.LogHandler:
zum Erfassen von Daten in einer Log-Datei
- medlog.db.store.MySqlHandler:
zur Datenspeicherung in einer MySQL-Datenbank

1.1.2 devices.xml

Die Konfigurationsdatei devices.xml beschreibt zum einen die angeschlossenen Geräte und zum anderen den Kontext, für den dieser Server eingesetzt und lizenziert wurde.

```
<devices>
  <license>
    <customer>Krankenhaus</customer>
    <department>Intensiv</department>
    <type>installation</type>
    <expirationDate>01.12.2010</expirationDate>
  </license>
  <medlog>
    <id>MedLog03</id>
    <control>true</control>
    <watchdog active="true">
      <udpPort>4130</udpPort>
      <loopDelay>10</loopDelay>
    </watchdog>
  </medlog>
  <device>
    <name>Ohmeda S5</name>
    <dongleId>10</dongleId>
    <deviceId>124551512424325</deviceId>
    <bedId>10</bedId>
    <className>
      medlog.device.driver.datex.OhmedaDriver
    </className>
    <active>true</active>
    <serial>
      <baudrate>19200</baudrate>
      <databits>8</databits>
      <stopbits>1</stopbits>
      <parity>none</parity>
      <flowcontrol>rts_cts</flowcontrol>
    </serial>
  </device>
</devices>
```

Eine ausführliche Erklärung der Tags folgt auf den nächsten Seiten.

Die devices.xml-Datei enthält viele geschachtelte Tags, die dazu dienen, ein neu an das MEDLOG-System angeschlossenes medizinisches Gerät zu konfigurieren.

• **<license>***Enthält geschachtelte Tags***</license>**

Einträge im Tag <license> dienen zur Identifikation des Inhabers einer MEDLOG-Lizenz zum Betreiben des Servers.

```
<license>
  <customer>Krankenhaus</customer>
  <department>Intensiv</department>
  <type>installation</type>
  <expirationDate>01.12.2010</expirationDate>
</license>
```

Die Tags in der Tabelle haben folgende Bedeutungen:

• **<customer>***Name des Kunden***</customer>**

Der Name des Kunden wird hier als String eingetragen, z.B. Krankenhausname, Dokumentationssystem-Hersteller.

• **<department>***Name der Abteilung, in der MEDLOG eingesetzt wird***</department>**

Hier wird die Abteilung eingetragen, in der das MEDLOG-System eingesetzt wird, z.B. Intensiv, Anästhesie.

• **<type>***Installationstyp***</type>**

Für die Vollversion wird der Wert "installation" eingetragen. Wird als <type> „ext-demo“ angegeben, dann wird der Server im Demo-Modus gestartet. Der Demo-Modus funktioniert nur sehr eingeschränkt, er kann und darf nicht als Datenlogger in einem realen Umfeld eingesetzt werden.

• **<expirationDate>***Ablaufdatum der Lizenz***</expirationDate>**

Ist ein <expirationDate> eingestellt, dann wird die Lizenz mit dem dort angegebenen Tag ablaufen, z.B. 01.12.2010 wie oben.



Hinweis: Wenn der Server mit einer gültigen Lizenz betrieben wird, dann dürfen die Felder innerhalb von <license> nicht verändert werden. Außerdem dürfen keine Treiber gelöscht oder neu hinzugefügt bzw. durch andere Treiber ausgetauscht werden.

• **<medlog>***Enthält geschachtelte Tags***<medlog>**

Die "Kindertags" von <medlog> dienen dazu, einen Server zu identifizieren und weitere nützliche Einstellungen vorzunehmen.

```
<medlog>
  <id>MedLog03</id>
  <control>true</control>
  <watchdog active="true">
    <udpPort>4130</udpPort>
    <loopDelay>10</loopDelay>
  </watchdog>
</medlog>
```

Die Tags in der Tabelle haben folgende Bedeutungen:

• **<id>***eindeutiger Name des MEDLOG-Servers***<id>**

Jeder Server verfügt über einen identifizierenden Namen, der über die externe Schnittstelle an das Dokumentationssystem weitergegeben wird. Der Name wird über das Tag <id> festgelegt, z.B. Medlog03.

• **<control>***"true" oder "false"***<control>**

Wird der Server als Applikation gestartet, öffnet sich eine einfache grafische Oberfläche, die anzeigt, welche Treiber erfolgreich gestartet werden konnten. Diese Anzeige kann über den Tag <control>false</control> abgeschaltet werden.

Für den Betrieb des Servers als Dienst sollte die Anzeige immer abgeschaltet sein.

• **<watchdog active="true">***enthält andere Tags***<watchdog>**

Um den Watchdog zu aktivieren, muss das Attribut „active“ des Tags <watchdog> auf „true“ gesetzt werden. Für die Kommunikation zwischen Watchdog und Server können optional der Port (<udpPort>) und ein Timeout (<loopDelay>) in Sekunden angegeben werden. Läuft das Timeout ab, ohne dass der Server dem Watchdog eine Mitteilung über seinen problemlosen Betrieb übergeben hat, dann stoppt der Watchdog den Server und startet ihn neu.

Ein Watchdog überwacht den Betriebsstatus des Rechners und wird in der Regel dort eingesetzt, wo es wirklich auf hohe Betriebssicherheit ankommt.

• **<udpPort>***Portnummer***<udpPort>**

Die Portnummer, über die die Kommunikation zwischen Watchdog und Server stattfindet, wird hier optional eingetragen. Eine Änderung des Ports für die Kommunikation des Watchdogs sollte nur dann vorgenommen werden, falls der Standardport von 4130 durch eine andere Software bereits belegt ist.

• **<loopDelay>***Timeout in Sekunden***</loopDelay>**

Hier wird optional ein Timeout in Sekunden angegeben. Innerhalb dieses Timeouts sollte die Kommunikation zwischen dem Server und dem Watchdog stattfinden, anderenfalls wird der Server nach dem Ablauf dieser Zeit neu gestartet.

• **<device>***Enthält geschachtelte Tags***</device>**

Einzelne Treiber werden durch die Angabe des Tags <device> konfiguriert.

Grundsätzlich verfügt ein Treiber über einen Namen, eine Device-ID, eine Bett-ID und eine verwendete Treiberklasse. Zusätzlich beinhaltet jeder Treiber eine Beschreibung der Parametereinstellungen für seine physikalische Schnittstelle.

```

<device>
  <name>Ohmeda S5</name>
  <dongleId>10</dongleId>
  <deviceId>124551512424325</deviceId>
  <bedId>10</bedId>
  <className>
    medlog.device.driver.datex.OhmedaDriver
  </className>
  <active>true</active>
  <serial>
    <baudrate>19200</baudrate>
    <databits>8</databits>
    <stopbits>1</stopbits>
    <parity>none</parity>
    <flowcontrol>rts_cts</flowcontrol>
  </serial>
</device>

```

Die Tags in der Tabelle haben folgende Bedeutungen:

• **<name>***Treibername bzw. Gerätename***</name>**

Für jeden Treiber sollte ein aussagekräftiger Name verwendet werden, aus dem deutlich hervorgeht, welches Gerät über diese Treiberinstanz angeschlossen ist.

• **<metaId>***Ganzzahliger Wert***</metaId>**

Die unverwechselbare Meta-ID des verwendeten Treibers wird in dieses Tag eingetragen.

• **<deviceId>***Zahlenwert als eindeutige ID des medizinischen Geräts***</deviceId>**

Jedes Gerät erhält eine eindeutige Nummer. Diese Nummer kann frei gewählt werden, jedoch sollte dabei beachtet werden, dass die Nummern über alle eingesetzten Server hinweg eindeutig sein sollten. Dadurch ist gewährleistet, dass ein Dokumentationssystem mit Hilfe der Device-ID einem Gerät eindeutig Parameter zuordnen kann.

• **<bedId>***ID des Bettes, wird meist vom Krankenhaus zugewiesen***</bedId>**

Jedem Gerät wird die Nummer eines Bettes zugewiesen (Bett-ID).

Mit Hilfe des Zeitstempel eines Parameters und der abgespeicherten Bett-ID kann ein Dokumentationssystem Patienten Parameter zuordnen.

• **<className>***Treibername***</className>**

Der Name des Treibers, der eingesetzt werden soll, wird in diesem Tag eingetragen.

• **<active>***"true" oder "false"***</active>**

Wird hier der Wert „true“ gesetzt, dann wird der Treiber beim Start des Servers gestartet. Soll der Treiber hingegen nicht verwendet werden, dann ist hier der Wert „false“ einzutragen.

• **<serial>***enthält weitere Tags***</serial>**

An dieser Stelle können Konfigurationen für die physikalische Schnittstelle folgen.

Im Beispiel liegt eine serielle Schnittstelle vor, die durch das Tag <serial> gekennzeichnet ist. Falls der Treiber über ein Netzwerk-Socket betrieben wird, muss an dieser Stelle das Tag <ipserver> verwendet werden (zusätzlich gibt es bei der Entwicklung die Möglichkeit Simulatoren zu konfigurieren. Das Tag dazu heißt <simulation>).

Die einzelnen Konfigurationen für die physikalischen Anschlussarten müssen speziell angepasst werden. Möglich ist zum einen die Konfiguration für die serielle Schnittstelle mit Direktanschluss oder unter Verwendung eines USB-Dongles, zum anderen gibt es die Anschlussmöglichkeit über ein Netzwerk. Zusätzlich kann ein Treiber auch als Simulator konfiguriert werden. Serielle Geräte werden über das Tag <serial> konfiguriert (siehe Beispiel in der Tabelle).

Im Folgenden soll nur die physikalische Konfiguration betrachtet werden:

```
<serial>  
  <baudrate>19200</baudrate>  
  <databits>8</databits>  
  <stopbits>1</stopbits>  
  <parity>none</parity>  
  <flowcontrol>rts_cts</flowcontrol>  
</serial>
```

• **<dongleId>***MEDLOG Device Adapter-Nr.***</dongleId>**

Für Geräte mit serielltem Anschluss, die über einen MEDLOG USB-Adapter angeschlossen werden, muss eine Dongle-ID eingetragen werden.

Diese Dongle-ID befindet sich auf einem Aufkleber auf dem Adapter und wird dort als „Seriennr.“ bezeichnet.

Die Tags in der Tabelle haben folgende Bedeutungen:

• **<baudrate>***Baudratenwert***</baudrate>**

Gibt die verwendete Baudrate des Gerätes an, oftmals können die medizinischen Geräte über verschiedene Baudraten angeschlossen werden (z.B. "9600", "19200").

• **<databits>***Datenbitanzahl***</databits>**

Anzahl von Datenbits für die serielle Schnittstelle, die für die Übertragung verwendet werden müssen (z.B. "7", "8").

• **<stopbits>***Stopbitanzahl***</stopbits>**

Anzahl von Stopbits für die serielle Schnittstelle, die für die Übertragung verwendet werden müssen (z.B. "1", "1.5", "2").

• **<parity>***"none", "even" oder "odd"***<parity>**

Die Parität ist ein einfacher Überprüfungsmechanismus, ob Daten richtig, ohne Verfälschung oder Verluste von medizinischen Geräten übertragen worden sind.

Als verwendete Parität können folgende Werte eingestellt werden:

- "none": es wird keine Parität verwendet
- "even": es wird gerade Parität verwendet
- "odd": es wird ungerade Parität verwendet

• **<flowcontrol>***"none", "rts_cts" oder "xon_xoff"***</flowcontrol>**

Über das <flowcontrol>-Tag können Einstellungen für die Flusskontrolle vorgenommen werden. Es können folgende Werte eingestellt werden:

- "none": keine Flusskontrolle
- "rts_cts": RTS/CTS eingeschaltet
- "xon_xoff": XON/XOFF eingeschaltet

Zur Verwendung der richtigen Parameter ist im Anhang für jedes Gerät ein Datenblatt mit möglichen Einstellungen enthalten. Alternativ kann auch die Dokumentation für das jeweilige medizinische Gerät verwendet werden. Dabei ist unbedingt zu beachten, dass die meisten Geräte entweder über Schalter oder über eine Menüsteuerung umgestellt werden können. Es ist daher notwendig, am Gerät festzustellen, welche Einstellungen dort vorgenommen worden sind.

In der Regel werden medizintechnische Geräte bereits in einem vorkonfigurierten Zustand ausgeliefert. Diese Vorkonfiguration wird in den Datenblättern anhand der jeweiligen Parameter hervorgehoben dargestellt.



Hinweis: Es muss darauf geachtet werden, dass beim Einsatz eines USB-Adapters die richtige Dongle-ID angegeben wird.

Neben dem Einsatz eines USB-Adapters kann auch eine herkömmliche serielle Schnittstelle am PC benutzt werden. Prinzipiell gelten die gleichen Einstellungen wie bei der Verwendung von USB-Adaptern, jedoch kommen zusätzliche Parameter hinzu: `<mode>` und `<interfaceName>`.

```
<serial>  
  <mode>COMM</mode>  
  <interfaceName>COM1</interfaceName>  
  <baudrate>19200</baudrate>  
  <databits>8</databits>  
  <stopbits>1</stopbits>  
  <parity>none</parity>  
  <flowcontrol>rts_cts</flowcontrol>  
</serial>
```

Die Tags in der Tabelle haben folgende Bedeutungen:

• **<mode>***verwendeter Modus***</mode>**

In diesem Tag wird der für die serielle Schnittstelle verwendete Modus eingetragen.

Wird dieses Tag weggelassen, so wird automatisch der USB-Modus für USB-Adapter verwendet, folgende Einstellungen sind zulässig:

- "USB": Modus für den Betrieb mit USB-Adaptern; hierfür muss eine gültige Dongle-ID gesetzt worden sein
- "COMM": Modus für herkömmliche RS232-Anschlüsse am Rechner (wird dieser Modus verwendet, so muss eine Schnittstellenbezeichnung mit `<interfaceName>` angegeben werden)
- "BLUETOOTH": Spezieller Modus für die Verwendung von kabellosen Bluetooth-Adaptern (der Bluetooth-Treiber erstellt einen virtuellen COMM-Port, der anschließend mit `<interfaceName>` angegeben werden muss)

• **<interfaceName>COM+Nr.</interfaceName>**

Für den Modus "COMM" muss zusätzlich angegeben werden, an welcher COMM-Schnittstelle das Gerät angeschlossen ist. Serielle Anschlüsse am Rechner verfügen nicht über „Plug'n'Play“-Eigenschaften und müssen daher genau spezifiziert werden (für in Windows-Betriebssysteme eingebaute Schnittstellen werden z.B. "COM1", "COM2" verwendet, Bluetooth-Treiber verwenden z.B. "COM7", "COM8").

Für Treiber basierend auf Netzwerkkommunikation muss anstelle eines <serial>-Tag ein <ipserver>-Tag angegeben werden. Eine vollständige Konfiguration für einen Treiber könnte folgendermaßen aussehen:

```
<device>
  <name>Ohmeda S5</name>
  <dongleId>10</dongleId>
  <deviceId>124551512424325</deviceId>
  <bedId>8</bedId>
  <className>
    medlog.device.driver.datex.OhmedaDriver
  </className>
  <active>>true</active>
  <ipserver>
    <host>192.168.1.99</host>
    <port>2403</port>
    <type>tcp</type>
  </ipserver>
</device>
```

Die fett gekennzeichneten Tags haben folgende Bedeutung:

• **<host>Rechnername bzw. IP-Adresse</host>**

Hier wird die IP oder der Hostname eingetragen.

Medizintechnische Geräte verfügen in der Regel über konfigurierbare IPs, die entsprechend der vorliegenden Infrastruktur konfiguriert sein müssen.

• **<port>Portnummer</port>**

Port, auf dem das medizintechnische Gerät horcht. Diese Angabe kann den Datenblättern im Anhang oder der Dokumentation des medizintechnischen Gerätes entnommen werden.

• **<type>"tcp" oder "udp"</type>**

Die Netzwerkkommunikation kann entweder auf "TCP" oder "UDP" basieren, die Einstellung hierfür muss den Datenblättern im Anhang entnommen werden. Mögliche Einstellungen sind "tcp" für TCP und "udp" für UDP.



Hinweis: Für die Konfiguration einer Reihe von Treibern müssen diese nacheinander hinzugefügt werden (siehe Tabelle auf der nächsten Seite).

Hinzufügen von mehreren Treibern:

```

<devices>

  <license>
    <customer>Krankenhaus</customer>
    <department>Intensiv</department>
    <type>customer</type>
    <expirationDate>01.12.2010</expirationDate>
  </license>

  <medlog>
    <id>MedLog03</id>
    <control>>true</control>
  </medlog>

  <device>
    <name>Ohmeda S5</name>
    <dongleId>10</dongleId>
    <deviceId>124551512424325</deviceId>
    <bedId>10</bedId>
    <className>
      medlog.device.driver.datex.OhmedaDriver
    </className>
    <active>>true</active>
    <serial>
      <baudrate>19200</baudrate>
      <databits>8</databits>
      <stopbits>1</stopbits>
      <parity>none</parity>
      <flowcontrol>rts_cts</flowcontrol>
    </serial>
  </device>

  <device>
    <name>Ohmeda S5</name>
    <dongleId>10</dongleId>
    <deviceId>124551512424325</deviceId>
    <bedId>8</bedId>
    <className>
      medlog.device.driver.datex.OhmedaDriver
    </className>
    <active>>true</active>
    <ipserver>
      <host>192.168.1.99</host>
      <port>2403</port>
      <type>tcp</type>
    </ipserver>
  </device>

  ...hier weitere Treiber hinzufügen...

</devices>

```


1.1.3 extern-interface.xml

Die Konfigurationsdatei extern-interface.xml steuert die Schnittstellen für das Dokumentationsmanagement.

Es können mehrere verschiedene externe Schnittstellen gleichzeitig genutzt werden, so dass für Dokumentationssysteme verschiedene Möglichkeiten zum Auslesen der Werte zur Verfügung stehen.

Falls spezielle externe Schnittstellen für spezielle Dokumentationssysteme verwendet werden sollen, dann muss die jeweils beiliegende Dokumentation beachtet werden (spezielle externe Schnittstellen können von der BMC auf besondere Bedürfnisse eines Dokumentationssystems hin entwickelt werden).

Folgende Schnittstellen sind bereits standardmäßig vorhanden:

Klassenname	Bedeutung
medlog.extern.DefaultExternManager	XML-Schnittstelle basierend auf HTTP
medlog.extern.hl7.trigger.HL7TriggerManager	Schnittstelle zur Verwendung des HL7-Sockets

Der oben aufgeführte Klassenname muss in der Konfigurationsdatei im Tag <interface> angegeben werden.

Es dürfen beliebige Schnittstellen hinzugefügt werden, jedoch muss dabei beachtet werden, dass von jeder Sorte nur jeweils ein Eintrag enthalten sein darf. Es ist nicht erlaubt, zwei gleiche Einträge in dieser Konfigurationsdatei vorzunehmen. Ein Beispiel für die Konfiguration sieht folgendermaßen aus:

```
<?xml version="1.0" encoding="UTF-8"?>
<externInterface>
  <interface class="medlog.extern.DefaultExternManager"/>
  <interface class="medlog.extern.hl7.trigger.HL7TriggerManager"/>
</externInterface>
```

In diesem Beispiel sind sowohl die Standardschnittstelle als auch das HL7-Socket aktiviert.

1.1.4 filter.xml

In der Konfigurationsdatei "filter.xml" können zusätzliche Filter gesetzt werden, die verschiedene Parameter abschalten. Abgeschaltete Parameter werden nicht mehr in der Datenbank gespeichert. Über einen Filter kann zum einen die Performanz des Servers verbessert und zum anderen mehr Übersichtlichkeit erreicht werden. Zusätzlich können vom Standard abweichende Mess-Einheiten gewählt und weitere Filtereigenschaften für den erweiterten Filtermodus konfiguriert werden (damit sich diese Filtereigenschaften auswirken, muss in devices.xml der Wert für „advancedFiltering“ auf „true“ gestellt sein).

Ist für eine Treiberinstanz kein Filter eingerichtet, dann werden alle Parameter in die Datenbank und in den internen Zwischenspeicher gesichert. Darüber hinaus findet keine Einheitenumwandlung statt, so dass generell alle Werte in der Standardeinheit vorliegen.

Soll für eine Treiberinstanz eine Filterung stattfinden, so muss in der Datei filter.xml für das jeweilige medizintechnische Gerät ein Eintrag einer Tabelle von Parametern vorgenommen werden. Alle in dieser Tabelle berücksichtigten Parameter werden gespeichert. Parameter, die nicht in der Tabelle aufgelistet sind, werden nicht gespeichert und können nicht mehr von einer externen Schnittstelle ausgelesen werden.

Zusätzlich lässt sich optional zu jedem Parameter eine abweichende Einheit angeben. Diese Einheit muss sich allerdings aus der Standardeinheit ableiten lassen.

Mit den Attributen „sampleRate“ und „tolerance“ lassen sich die Parameter für den erweiterten Filtermodus konfigurieren.

Die Samplerate ist in Sekunden anzugeben und beschreibt eine Mindestdauer für diesen Parameter: Zeit, die verstrichen sein muss, bevor ein weiterer Wert gespeichert wird.

Mit der Angabe „tolerance“ kann man von der Samplerate in dem Fall abweichen, in dem ein Wert sich gegenüber dem vorangegangenen Wert überdurchschnittlich verändert hat. Ist diese Änderung größer als in „tolerance“ konfiguriert, so wird der Wert auch dann abgespeichert, wenn die über die Samplerate eingestellte Zeitspanne noch gar nicht abgelaufen ist. Dadurch können Kurvensprünge und besondere Ereignisse auch bei langen Abtastzeiträumen berücksichtigt werden. Die Eingabegröße von „tolerance“ ist abhängig von der Einheit des Wertes aus dem Standard. Wird keine Samplerate angegeben, so liegt diese standardmäßig bei 300, der Standardwert für „tolerance“ ist 5.

Ein Beispiel für einen Filter sieht folgendermaßen aus:

```
<filter>
  <device deviceId="2534654575756424626">
    <parameter name="RESVENASB" targetUnit="ml"/>
    <parameter name="RESVENATE" targetUnit="ml/min"/>
    <parameter name="RESVENFAB" targetUnit="1/h" sampleRate="40"/>
    <parameter name="RESVENSBE"/>
  </device>
  <device deviceId="493074447221741226">
    <parameter name="INJPERINV"/>
    <parameter name="INJPERINS" targetUnit="l" tolerance="3.0"/>
    <parameter name="INJPERFOK" sampleRate="40" tolerance="3.0"/>
    <parameter name="INJPERINZ"/>
    <parameter name="INJPERBVL" targetUnit="l"/>
  </device>
  <device deviceId="123423463523656582">
    <parameter name="RESVENPLP" targetUnit="1/min"/>
  </device>
  <device deviceId="545763454766784563">
    <parameter name="HAMRHYFAB" targetUnit="1/h"/>
    <parameter name="RESANAHIN"/>
    <parameter name="RESANAEEX"/>
    <parameter name="RESANAEAG"/>
  </device>
</filter>
```

In diesem Beispiel wurden Filter für vier verschiedene Geräte eingerichtet. Die Geräte werden von den Treibern mit den entsprechenden Device-IDs benannt. Diese Device-IDs müssen mit denen aus der devices.xml übereinstimmen. Für jedes Gerät wird ein <device>-Tag-Block erstellt:

```
<device deviceId="2534654575756424626">
  <parameter name="RESVENASB" targetUnit="ml"/>
  <parameter name="RESVENATE" targetUnit="ml/min"/>
  <parameter name="RESVENFAB" targetUnit="1/h"/>
  <parameter name="RESVENSBE"/>
</device>
```

Für das Gerät mit der Device-ID „2534654575756424626“ wurden vier Parameter ausgewählt. Alle **weiteren** gelieferten Parameter werden verworfen und nicht gespeichert. Für jeden Parameter werden der Name und die Zieleinheit angegeben. Die Angabe der Zieleinheit ist optional und muss nur dann erfolgen, wenn eine von der Standardeinheit abweichende Zieleinheit gewünscht wird. Als Name wird der Kategorienname gewählt.



Hinweis: Es muss auf die richtige Schreibweise geachtet werden. Der 9-stellige Kategorienname muss richtig geschrieben sein. Der Eintrag für die Einheit muss so eingepflegt werden, wie er im Anhang dargestellt ist.

1.1.5 log.xml

Die Einstellungen für die Konfigurationsdatei "log.xml" beziehen sich auf den vom Server verwendeten Logging-Mechanismus, dafür wird Log4J¹ von Apache Jakarta verwendet.

Die ausgelieferte, vorkonfigurierte log.xml erzeugt im Verzeichnis „logs“ verschiedene Log-Dateien:

- Die Datei "standard.log" enthält alle maßgeblichen Log-Einträge des Servers,
- In "system.log" werden Systemnachrichten abgespeichert.
- Für die externen Schnittstellen ist "extern.log" zuständig
- Der Watchdog legt die Dateien "watch.log" und "watch_event.log" an

Log4J unterstützt mehrere Log-Level. Die Voreinstellungen sehen vor, dass Nachrichten ab dem Log-Level "Warnung" gespeichert werden. Soll ein niedrigeres Log-Level verwendet werden, dann kann die Datei log.xml entsprechend angepasst werden. Es wird dazu auf die entsprechende Dokumentation von Log4J verwiesen.

Desweiteren können in der Datei die Benachrichtigungen des Administrators über Lizenzinformationen sowie über fatale Fehler im MEDLOG-Server aktiviert werden.

Dazu werden die folgenden Einträge der Datei log.xml hinzugefügt:

```
<appender name="AdminNotification" class="org.apache.log4j.RollingFileAppender">
  <param name="SMTPHost" value="mailserver"/>
  <param name="From" value="mail@domain.de"/>
  <param name="To" value="mail@domain.de"/>
  <param name="Subject" value="[App Alert]"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d\t%p\t(%F:%L) %m%n"/>
  </layout>
</appender>
```

Dieser Eintrag erzeugt einen so genannten Appender für das Versenden von Nachrichten an den Administrator.

- `<param name="From" value="mail@domain.de"/>`
legt den Absender der Nachrichten fest.
- `<param name="To" value="mail@domain.de"/>`
legt den Empfänger der Nachrichten fest.

¹ <http://logging.apache.org/log4j/docs/index.html>

Des Weiteren kann durch die folgenden Einträge bestimmt werden, welche Art von Nachrichten über den oben erzeugten Appender versendet werden.

Für alle speziellen Administrationsnachrichten (z.B. Lizenzinformationen, Ablaufdatum):

```
<logger name="admin.logger.notification" additivity="false">
  <level value="debug"/>
  <appender-ref ref="AdminNotification"/>
</logger>
```

Für fatale Fehler im Server:

```
<logger name="admin.logger.notification" additivity="true">
  <level value="fatal"/>
  <appender-ref ref="AdminNotification"/>
</logger>
```

1.1.6 servlet.xml

Die Konfigurationsdatei "servlet.xml" konfiguriert das Verhalten der XML-Schnittstelle, der externen Standardschnittstelle.

```
<medlog>
  <extern>
    <port>8080</port>
    <parameterLimit>2500</parameterLimit>
    <characterEncoding>ISO-8859-15</characterEncoding>
    <latestValuesCacheMaxTime>10000</latestValuesCacheMaxTime>
    <useStylesheets>true</useStylesheets>
    <users>
      <medlogUser userName="medlog_ext" userPassword="medl0g"/>
      <medlogAdmin userName="medlog_admin" userPassword="medl0g"/>
    </users>
  </extern>
</medlog>
```

Die Tags haben folgende Bedeutung:

- **<port>XML-Schnittstelle Port "8080"</port>**

Voreingestellt als Port für die XML-Schnittstelle ist Port 8080. Sollen die Daten auf einem davon abweichenden Port ausgeliefert werden, so muss für das Tag <port> ein anderer Wert eingetragen werden. Dies ist notwendig, wenn beispielsweise bereits ein anderer Server diesen Port verwendet.

Falls auf dem Computer nur MEDLOG installiert ist, und der Port 80 nicht durch einen Webserver belegt wurde, dann kann auch dieser Port verwendet werden. Werden Anfragen über einen Browser gestellt, dann kann auf die Angabe des Ports verzichtet werden, weil der Browser ohnehin standardmäßig Port 80 anfragt.

- **<parameterLimit>Maximalparameteranzahl</parameterLimit>**

Das Tag <parameterLimit> gibt die Maximalanzahl von Parametern an, die bei einer Anfrage übertragen werden. Umfasst eine Anfrage an die externe Schnittstelle einen zu großen Zeitraum, so dass die Parameteranzahl über diesem Limit liegt, dann wird die Anfrage verworfen und nicht beantwortet. Eine Limitierung der Parameterübertragung kann dazu dienen, die Antwortzeiten für die Parameteranfragen gering zu halten.

- **<characterEncoding>ISO-8859-15</characterEncoding>**

Die externe Schnittstelle erzeugt als Antwort auf eine Datenanfrage XML-Nachrichten. Diese werden mit einem bestimmten Character-Encoding erzeugt. Das angegebene Beispiel verwendet z.B. ISO-8859-15. Diese Angabe ist wichtig für Dokumentationssysteme, die nur mit bestimmten Encodings arbeiten können.

Werden Anfragen eventuell über einen Browser abgesetzt, so kann es zu Darstellungsfehlern im Browser kommen, wenn dieser nicht die entsprechende Dekodierung unterstützt.

Gebräuchliche Encodings sind in der nachfolgenden Tabelle auf der nächsten Seite angegeben.

Encoding	Bedeutung
US-ASCII	Seven-bit ASCII, auch ISO646-US oder der "Basic Latin block of the Unicode character set"
ISO-8859-1	ISO Latin Alphabet No. 1, auch ISO-LATIN-1
ISO-8859-15	ISO Latin Alphabet No. 9
UTF-8	Eight-bit Unicode Transformation Format
UTF-16BE	Sixteen-bit Unicode Transformation Format, Byte-Ordnung: Big-endian
UTF-16LE	Sixteen-bit Unicode Transformation Format, Byte-Ordnung: Little-endian
UTF-16	Sixteen-bit Unicode Transformation Format, Byte-Ordnung über eine erforderliche Startmarkierung (Ordnung bei Eingabe akzeptiert, Big-endian als Output verwendet)

• **<latestValuesCacheMaxTime>max.Verweildauer</latestValuesCacheMaxTime>**

Das Tag <latestValuesCacheMaxTime> gibt die maximale Verweildauer für Parameter im Zwischenspeicher an. Die Angabe wird in Millisekunden vorgenommen. Überschreitet ein Parameter die maximal angegebene Verweildauer, so wird er aus dem Zwischenspeicher gelöscht.

• **<useStylesheets>"true" oder "false"</useStylesheets>**

Mit der Einstellung <useStylesheets> kann das Ausliefern von XSL-Dateien an Browser aktiviert werden. Ist sie über den Eintrag „true“ eingeschaltet, dann wird anstelle des XMLs die grafische Darstellung im Browser angezeigt, sofern der verwendete Browser XSLT unterstützt.

• **<users>Enthält geschachtelte Tags</users>**

Für den Datenzugriff per HTTP ist eine Authentifizierung notwendig.

Es gibt ein Benutzerkonto für administrative Zwecke <medlogAdmin> und eines für die Abfrage von Daten <medlogUser>.

Für beide Konten müssen jeweils ein Name und ein Passwort angegeben werden. Diese müssen entweder vom Benutzer beim Aufruf der Daten über einen Browser in einen Dialog eingetragen werden, oder das Dokumentationssystem berücksichtigt bei der HTTP-Anfrage eine entsprechende Basic-Authentifizierung (dafür müssen Name und Passwort mit einem Doppelpunkt getrennt als Base64-codierte Zeichenkette mit einem vorangehenden „Basic“ angegeben werden und unter „Property Authorization“ abgelegt sein, z.B. Basic <name:passwort>base64).

• **<medlogUser userName="medlog_ext" userPassword="medl0g"/>**

Ein Benutzerkonto für die Abfrage von Daten.

• **<medlogAdmin userName="medlog_admin" userPassword="medl0g"/>**

Ein Benutzerkonto für administrative Zwecke.

1.1.7 hl7.xml

Die Konfigurationsdatei "hl7.xml" ist für die HL7-Schnittstelle zuständig.

```
<hl7>  
  <triggerPort>7998</triggerPort>  
  <waitingPeriod>10</waitingPeriod>  
</hl7>
```

• **<triggerPort>***Standardport "7998"***</triggerPort>**

Über die Datei "hl7.xml" kann ein abweichender Port für das HL7-Socket eingestellt werden. Der Standardport für das HL7-Socket liegt bei 7998. Um einen anderen Port einzustellen, muss die Datei hl7.xml angepasst werden.

• **<waitingPeriod>***Zeitspanne in Sekunden***</waitingPeriod>**

Mit der Einstellung für „waitingPeriod“ wird die Zeitspanne in Sekunden eingestellt, die zwischen zwei HL7-Nachrichten vergeht.

1.1.8 sslserver.xml

Damit der Administrations-Editor Zugriff auf einen MEDLOG-Server bekommt, muss zunächst eine SSL-Verbindung aufgebaut werden. Dazu wird eine sslserver.xml-Konfigurationsdatei benötigt.

```
<sslserver>
  <keystore>keystore</keystore>
  <keystorePassword>password</keystorePassword>
  <port>4120</port>
</sslserver>
```

- **<sslserver>** Enthält geschachtelte Tags **</sslserver>**

Die Tags `<sslserver>` ermöglichen über eine SSL-Verbindung Zugriff auf einen MEDLOG-Server.

- **<keystore>**`keystore`**</keystore>**

Das Tag `<keystore>` gibt die Datei des Keystores an. In diesem Keystore ist ein Zertifikat für den Server enthalten. Der Administrations-Editor verfügt über einen Truststore, in dem ebenfalls das Zertifikat vorhanden ist. Die Erzeugung von Keystore- und Truststore-Dateien wird im Folgenden erläutert.

- **<keystorePassword>**`password`**</keystorePassword>**

Für das Keystore wurde ein Passwort angegeben. Dieses Passwort muss auch im Tag `<keystorePassword>` angegeben werden.

- **<port>**`password`**</port>**

Der Port für die SSL-Übertragung wird im Tag `<port>` angegeben. Es wird empfohlen, die Einstellung auf dem Standardport von 4120 zu belassen.

1.1.9 serverlist.xml

Beim Einsatz von mehreren MEDLOG-Systemen im Verbund eines verteilten Systems müssen medizinische Parameter zwischen den Servern ausgetauscht werden. Dazu ist es notwendig, eine logische Baumstruktur zwischen den MEDLOG-Systemen herzustellen.

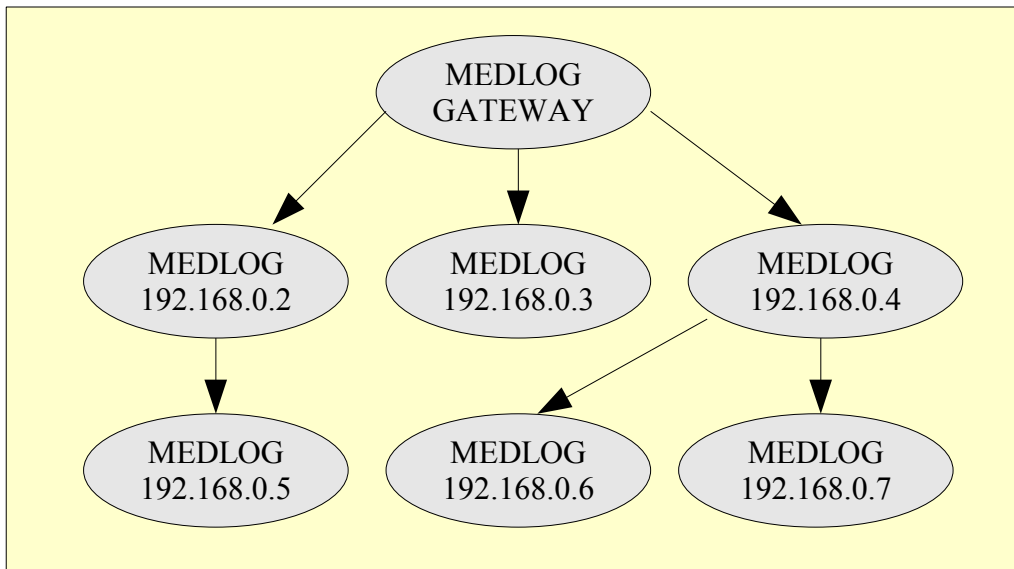


Abbildung 1: Darstellung mehrerer MEDLOG-Systeme in der Anordnung eines logischen Baums für ein verteiltes System

In Abbildung 1 wird ein Beispiel mit sieben MEDLOG-Installationen dargestellt. Eine dieser Installationen fungiert als Gateway-Server, der von einem Dokumentationssystem angesprochen wird. Die übrigen Server sind dem Gateway direkt oder indirekt unterstellt.

Der Gateway-Server spricht die Server mit den IP-Adressen 192.168.0.2 bis 192.168.0.4 direkt an. Dazu müssen in der Konfigurationsdatei <serverlist.xml> alle diese IPs angegeben worden sein. Die Konfiguration für das Gateway aus diesem Beispiel würde folgendermaßen aussehen:

```
<medlog>
  <serverList>
    <server host="192.168.0.2" port="8080" userName="medlog_ext"
  userPassword="medlog"/>
    <server host="192.168.0.3" port="8080" userName="medlog_ext"
  userPassword="medlog"/>
    <server host="192.168.0.4" port="8080" userName="medlog_ext"
  userPassword="medlog"/>
  </serverList>
  <networkTimeout>5000</networkTimeout>
</medlog>
```

Die Tags haben folgende Bedeutung:

• **<serverList>** Enthält geschachtelte Tags **</serverList>**

Dieses Tag fasst mehrere Server-Tags zusammen.

• **<server host="192.168.0.2" port="8080" userName="medlog_ext" userPassword="medlog"/>**

Für alle Server wird angenommen, dass als XML-Schnittstelle der Port 8080 auf den Servern konfiguriert wurde. Als Benutzerdaten werden im Beispiel Standardname und -passwort verwendet. Sobald die jeweiligen Server über neue Benutzerdaten verfügen, müssen auch die Angaben an dieser Stelle nachgepflegt werden (die Angaben für die Benutzer werden in der Konfigurationsdatei „servlet.xml“ vorgenommen).

Der Computer mit der IP 192.168.0.4 verfügt ebenfalls über zwei ihm untergeordnete Rechner, daher benötigt er ebenfalls lokal eine serverlist.xml. Wird ein Rechner im Standalone-Betrieb benutzt, oder hat er keine weiteren Rechner unter sich in der Baumstruktur, wie im Beispiel der Rechner mit der IP 192.168.0.5, dann wird eine leere Liste angegeben.

• **<networkTimeout>5000</networkTimeout>**

Über den Eintrag „networkTimeout“ kann eingestellt werden, wie lange der MEDLOG-Server maximal auf eine Antwort von einem anderen Server wartet.

Ist ein Server abgeschaltet, so kann keine Kommunikation zu diesem Rechner aufgebaut werden. Nach einer bestimmten eingestellten Wartezeit, im Beispiel 5 Sekunden, reagiert das System, und der Rechner wird ausgelassen. Wird der Wert der Wartezeit zu niedrig angegeben, kann unter Umständen in Netzwerken mit hoher Belastung eine Kommunikation nicht hergestellt werden, obwohl ein Kommunikationspartner aktiv ist. Ist der Wert zu hoch, dann erhöht sich auch die Antwortzeit des MEDLOG Netzwerkes, falls ein Rechner nicht erreichbar ist.

Um die Antwortzeiten möglichst niedrig zu halten, werden Anfragen an die untergeordneten Server in der Baumstruktur parallel durchgeführt, sie sind daher maßgeblich vom langsamsten Rechner einer Ebene abhängig. Die Antwortzeit insgesamt bildet sich aus der Summe der Antwortzeiten jeder Ebene.

Im Beispiel bilden die Rechner mit den IPs 192.168.0.2-4 und 192.168.0.5-7 jeweils eine Ebene.

1.1.10 realtime.xml

Die Konfigurationsdatei "realtime.xml" konfiguriert das Verhalten beim Echtzeitexport, sofern dieser in der Datei extern-interface.xml aktiviert ist.

• **<hostName>localhost</hostName>**

Hostname oder IP-Adresse (optional – falls angegeben, fungiert MEDLOG als Client) müssen hier angegeben werden.

• **<serverPort>9001</serverPort>**

Verbindungsport, die Standardeinstellung ist 9001.

• **<levelName>plain</levelName>**

Ebene für den Datenbankmechanismus, die vom StoreHandler verwendet wird. Da dieser automatisch registriert ist, muss die Registrierung nicht manuell als Datenbankregistrierung erfolgen.

• **<realtimeClientHandler>medlog.extern.realtime.handler.BinaryRealtimeClientHandler</realtimeClientHandler>**

Klasse der RealtimeClientHandler-Implementierung.

• **<liveTickInterval>30</liveTickInterval>**

Zeitintervall (in Sekunden) zwischen Statusnachrichten aller Geräte, die mit MEDLOG verbunden sind (nur für Binärschnittstellen).

• **<reconnectInterval>60</reconnectInterval>**

Zeitintervall (in Sekunden) zwischen den Wiederverbindungsversuchen von MEDLOG nach dem Verlust einer Verbindung (nur für Binärschnittstellen).

• **<resendLabValuesMinutes>1</resendLabValuesMinutes>**

Zeitintervall (in Minuten), in dem MEDLOG Werte von Laborgeräten erneut sendet (nur für Binärschnittstellen).

• **<maxAgeLabValuesMinutes>360</maxAgeLabValuesMinutes>**

Time-out-Intervall (in Minuten) für Werte von Laborgeräten, die nicht zu einem Empfänger gesendet wurden (nur für Binärschnittstellen).

• **<workingMode>** *“async” oder “sync”* **</workingMode>**

Dieses Tag definiert den Arbeitsmodus.

"async" – Asynchrone Verarbeitung, keine Blockierung der Schnittstelle nach der Übertragung von Laborgerätedaten. Für Laborgerätedaten wird immer ein Backup erstellt. Die Backup-Datei wird gelöscht, sobald eine Bestätigung vom externen System empfangen wird.

"sync" – Synchrone Verarbeitung, wartet für eine einstellbare Zeit nach der Übertragung von Laborgerätedaten (z.B. BGA = Blutgasanalyse) auf Bestätigung. Wird keine Bestätigung empfangen, werden die Daten in eine Backup-Datei gespeichert (nur für Binärschnittstellen).

• **<responseTimeout>** *100* **</responseTimeout>**

Ist der Arbeitsmodus auf "sync" gesetzt, bestimmt das "responseTimeout" die Zeit, die das System auf eine Antwort vom Dokumentationssystem wartet (nur für Binärschnittstellen).

1.2 Keystore und Truststore

Für den gesicherten Zugriff auf die Konfiguration des Servers können ein eigenes Keystore für den Server und ein Truststore für den Editor erzeugt werden. Keystore und Truststore werden in den Dateien „keystore“ (im Verzeichnis conf/server) und „truststore“ (im Verzeichnis conf/editor) gespeichert; sslserver.xml und sslclient.xml von Server und Administrations-Editor müssen entsprechend auf die Dateien verweisen.

Für die Erzeugung der Keystore- und Truststore-Dateien unter Windows kann das Tool "keytool" verwendet werden, welches dem Java JDK¹ beiliegt.

Folgende Schritte zur Erzeugung der Dateien sind notwendig:

- Öffnen einer Kommandozeile (Start -> Ausführen -> cmd)
- Aufruf von "keytool - genkey - keystore keystore - storepass password - validity 360" (keystore ist der Name des Keystores, password ist das gewählte Passwort, 360 ist die Anzahl Tage, die dieses Zertifikat gültig ist.)
Die gestellten Fragen müssen beantwortet werden, bei der Frage nach einem Passwort für <mykey> Return drücken.
- Aufruf von "keytool - list - v - keystore keystore"
Passwort eingeben und Ausgaben verifizieren.
- Aufruf von "keytool - export - alias mykey - keystore keystore - rfv - file key.cer"
Passwort eingeben. Eine Datei key.cer mit dem Zertifikat wurde angelegt.
- Aufruf von "keytool - import - alias mykey - file key.cer - keystore truststore" (importiert das Zertifikat in die Datei "truststore")
- Aufruf von "keytool - list - v - keystore truststore"
Passwort eingeben und Ausgaben verifizieren.
- Die Datei „keystore“ in das Konfigurationsverzeichnis des Servers kopieren und die Datei „truststore“ in das Konfigurationsverzeichnis des Administrations-Editors kopieren.

¹ www.java.sun.com

1.3 Lizenzdateien

Für den Betrieb eines MEDLOG-Servers sind zwei Lizenzdateien erforderlich. Die Datei "medlog-license.dat" enthält die Lizenz für einen Rechner. Für jeden Rechner wird eine eindeutige Seriennummer ermittelt, die für diese Lizenzdatei vorhanden sein muss. Die Lizenzdatei wird von BMC erstellt und ausgeliefert; zuvor muss die Seriennummer des Rechners ermittelt werden. Die Seriennummer kann beispielsweise über den Administrations-Editor auf der Lizenz-Seite ausgelesen werden, oder sie ist nach einem Start des Servers in der Log-Datei standard.log zu finden.

Zwei weitere Möglichkeiten, die Seriennummer ohne eine vorherige Installation anzeigen zu lassen: Den Server als Applikation starten und in der Anzeige die Seriennummer ablesen, oder den Start des Servers mit einem zusätzlichen Parameter durchführen. Dazu wird der Server folgendermaßen gestartet:

- Kommandozeile öffnen (Start -> Ausführen -> cmd)
- Mit dem Befehl "cd" in das Installationsverzeichnis des MEDLOG-Applikationsservers wechseln (z.B. "cd c:\Programme\bmc")
- Server mit folgendem Aufruf starten: "MedLogServer.exe -serial"

Die zweite Lizenzdatei "driver-license.dat" enthält die Lizenz für alle freigeschalteten Treiberinstanzen. Für eine "devices.xml" kann von der BMC eine Lizenzdatei erzeugt werden, die dann die Treiber und erweiterten Funktionalitäten des Servers für einen ordnungsgemäßen Betrieb freischaltet.

Es muss dabei beachtet werden, dass in der "devices.xml" keine Treiber hinzugefügt oder gelöscht werden dürfen. Zudem dürfen die lizenzbezogenen Daten nicht verändert werden.

Die Lizenzdateien werden von BMC erstellt und müssen in das Konfigurationsverzeichnis des Servers ("conf/server") kopiert werden.

1.4 Demo-Installation

Der MEDLOG-Server kann zu Demonstrationszwecken im Demo-Modus betrieben werden.

Für den Demo-Betrieb ist es erforderlich, dass keine Lizenzdateien installiert sind und dass ein Eintrag in der "devices.xml" vorgenommen wird. Im Tag <type> muss als Wert "ext-type" eingestellt werden, wie im nachfolgenden Beispiel verdeutlicht:

```
<devices>

  <license>
    <customer>Krankenhaus</customer>
    <department>Intensiv</department>
    <type>ext-type</type>
    <expirationDate>01.12.2010</expirationDate>
  </license>

  ...

</devices>
```

In der Demo-Version von MEDLOG sind einige Funktionen eingeschränkt:

- Beschränkung auf maximal vier Treiber
- Kein Datenbankspeicher
- Keine Filterung der Parameter
- 30 Tage Probelaufzeit

1.5 Alternative Konfiguration

Falls eine alternative Konfiguration verwendet werden soll, die nicht im Standardverzeichnis des Servers gespeichert ist, so kann dies über einen modifizierten Start des Servers durchgeführt werden. Der Start muss um den Parameter "-conf" ergänzt werden, gefolgt vom Speicherort der alternativen Konfiguration, z.B. "MedLogServer.exe -conf c:\temp\conf".