

SystemTek

2016

SystemTek GWS  
Kommunikationslösungen  
LDT v2.x-Extension



Reiner Neubert

SysTek GmbH

03.02.2016

## Inhalt

SysTek Kommunikationslösungen .....	2
Kurzbeschreibung der Verarbeitung von LDT Nachrichten .....	2
SysTek LDT-Extension für Mirth Connect .....	3
Technische Voraussetzungen zur Installation der SysTek LDT v2.x-Extension .....	3
Update auf eine aktuelle Mirth Connect Version .....	3
Installation der SysTek LDT v2.x-Extension .....	3
Schritt 1 der Installation: .....	4
Schritt 2 der Installation: .....	5
Problemlösungen für fehlerhafte eine Installation .....	6
Grundsätzliche Verarbeitung von Nachrichten in Mirth Connect .....	7
Nutzung der SysTek LDT v2.x-Extension .....	8
Inbound Properties .....	9
Serialization .....	9
Batch .....	11
Outbound Properties .....	12
Deserialization .....	12
Template Serialization .....	13
Filter, Transformation, Mappings und Templates .....	13
Templates .....	13
Filter .....	15
Transformation .....	<b>Fehler! Textmarke nicht definiert.</b>
Mappings .....	<b>Fehler! Textmarke nicht definiert.</b>
Fazit .....	<b>Fehler! Textmarke nicht definiert.</b>

## SysTek Kommunikationslösungen

Die SysTek GmbH ist spezialisiert auf die Planung, Realisierung und Erstellung von sicherer elektronischer komplexer Nachrichtenkommunikation in medizinischen Netzwerken. Wir übernehmen für Sie die Ausarbeitung eines optimal passenden Workflows, die Realisierung und langfristige umfassende Betreuung und Wartung Ihrer medizinischen Kommunikationslösung auf der Basis von Mirth Connect™.



Durch unsere langjährigen Erfahrungen im Bereich der Entwicklung von Software für medizinische Laboratorien wie LIS-Systemen (MultiLAB, proLAB), LIS-Komponenten (Sysmex TWIST, OLA Manager, SysTek Online Control Server, Probenarchivierungssystemen etc.), mobiler Befundauskunftssysteme (LABscope-APP) u.v.m. steht Ihnen mit SysTek ein kompetenter Partner zur Seite.

## Kurzbeschreibung der Verarbeitung von LDT Nachrichten

Die LDT-Datensatzbeschreibung (Labordatenträger) ermöglicht die Übertragung von Aufträgen und Befunden zwischen Einsendepatienten und Arztpraxen. Es handelt sich um eine Textdatei die in jeder Zeile eine LDT Feldinformation enthält.

```
01380008230
014810000326
0299901*Information zu 8230
0199212LDT1014.01
0180201198492200
0170203LG-Labor
0180212779866301
0180211Dr. Meyer
0180222001234566
0260205xxxxxxxxxxxxxxxx 88
```

Jedes Feld setzt sich aus 4 Komponenten zusammen:

019 9212 LDT1014.01 CR+LF

Feldname	Länge des Feldes	Bedeutung des Feldes
Länge	3 Bytes	Feldlänge in Bytes
Feldkennung	4 Bytes	Feldkennung
Inhalt	Variable	Inhalt des Feldes
Ende	2 Bytes	CR+LF (chr(13)+chr(10))

Komplette Nachrichten, Position, Reihenfolge und Inhalt der LDT Felder sind durch die KBV (Kassenärztliche Bundesvereinigung) vorgeschrieben und definiert. Es existieren genaueste Dokumentationen und Standards. Ein Softwareprodukt muss zertifiziert sein um LDT Dateien zu verarbeiten und zu erstellen. SysTek hat alle Definitionen der KBV in XSD Schemas und XML Dateien definiert.

## SysTek LDT v2.x Extension für Mirth Connect

Mit der SysTek LDT-Extension wird Mirth Connect mit einem Java Plug-In um den Datentyp LDT v2.x erweitert. Nun können mit Mirth Connect LDT Dateien empfangen, validiert, transformiert und versandt werden. Der LDT Datentyp wird zur Bearbeitung in ein XML Format gewandelt und steht somit zur umfangreichen Bearbeitung in Mirth Connect bereit.

## Technische Voraussetzungen zur Installation der SysTek LDT v2.x-Extension

Eine Mirth Connect Extension ist abhängig von der installierten Mirth Version. Beide Versionsstände müssen übereinstimmen. Um das zu gewährleisten verfügt jede Extension über ein XML File mit dem Namen „plugin.xml“. Bei der Installation einer Extension wird diese Datei von Mirth Connect überprüft und bei Ungleichheit abgelehnt.

Aktuell muss für die SysTek LDT2.x-Extension eine Mirth Connect Version  $\geq$  3.3.0 installiert sein.

Sie können ihre installierte Version im Mirth Connect Administrator über den Menüpunkt „About Mirth Connect“ abfragen. Hier sehen Sie ein Beispiel:



Ältere Versionen von Mirth Connect werden durch SysTek Extensions nicht unterstützt.

## Update auf eine aktuelle Mirth Connect Version

SysTek führt gerne für Sie ein Update für auf die aktuellste Mirth Connect Version durch.

## Installation der SysTek LDT v2.x-Extension

Die Mirth Connect Extension wird in Form eine ZIP Datei geliefert. Diese ZIP Datei enthält alle Bestandteile einer Mirth Connect Extension. Diese kann im Mirth Connect Administrator installiert werden. Eine ältere SysTek LDT2.x-Extension muss vorher deinstalliert werden.

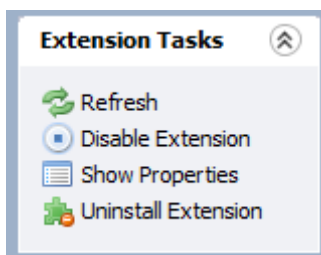
## Schritt 1 der Installation:

Wählen Sie im Mirth Connect Administrator den Menüpunkt „Extensions“. In der Liste „Installed Plugins“ können Sie die aktuell installierten Plug-Ins einsehen und warten.

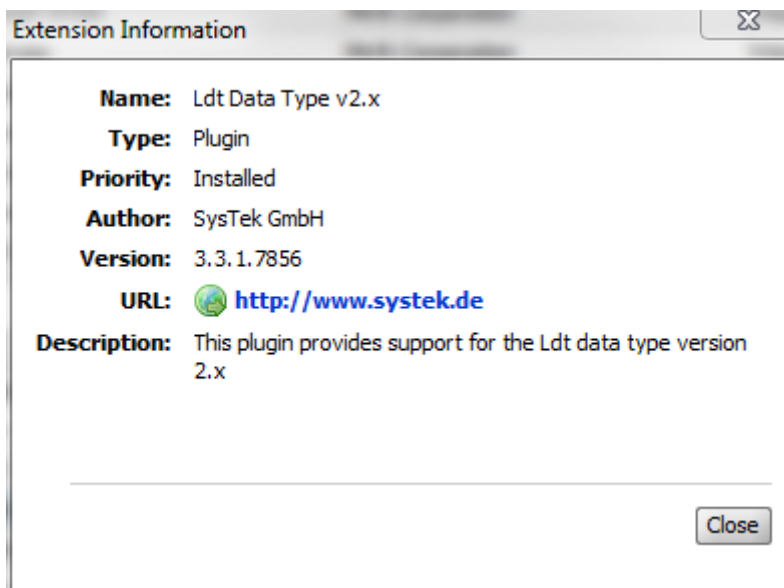
Hier eine Beispielansicht:

Installed Plugins				
Status	Name	Author	URL	Version
Enabled	Message Tree	SysTek GmbH	<a href="http://www.systemek.de">http://www.systemek.de</a>	3.3.1.7856
Enabled	History	SysTek GmbH	<a href="http://www.systemek.de">http://www.systemek.de</a>	3.3.0.7856
Enabled	Report	SysTek GmbH	<a href="http://www.systemek.de">http://www.systemek.de</a>	3.3.0.7801
Enabled	Ldt Data Type v2.x	SysTek GmbH	<a href="http://www.systemek.de">http://www.systemek.de</a>	3.3.1.7856
Enabled	Ldt Data Type v3.x	SysTek GmbH	<a href="http://www.systemek.de">http://www.systemek.de</a>	3.3.1.7856
Enabled	Server Log	Mirth Corporation	<a href="http://www.mirthcorp.com">http://www.mirthcorp.com</a>	3.3.1.7856

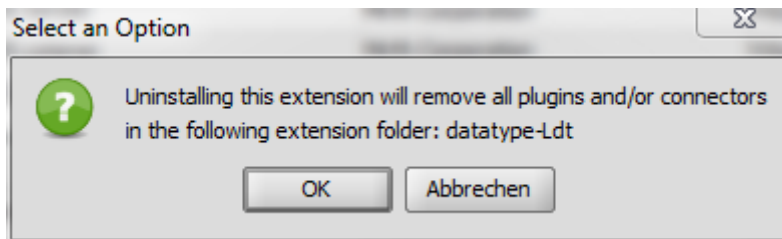
Sie können nun folgende Aufgaben durchführen:



Lassen Sie sich die Eigenschaften der Extension mit dem Task „Show Properties“ anzeigen:



In diesem Fall müssen Sie die alte Version der SysTek LDT2.x-Extension vorher deinstallieren. Wählen Sie dazu den Task „Uninstall Extension“. Bestätigen Sie die anschließende Meldung mit „OK“.

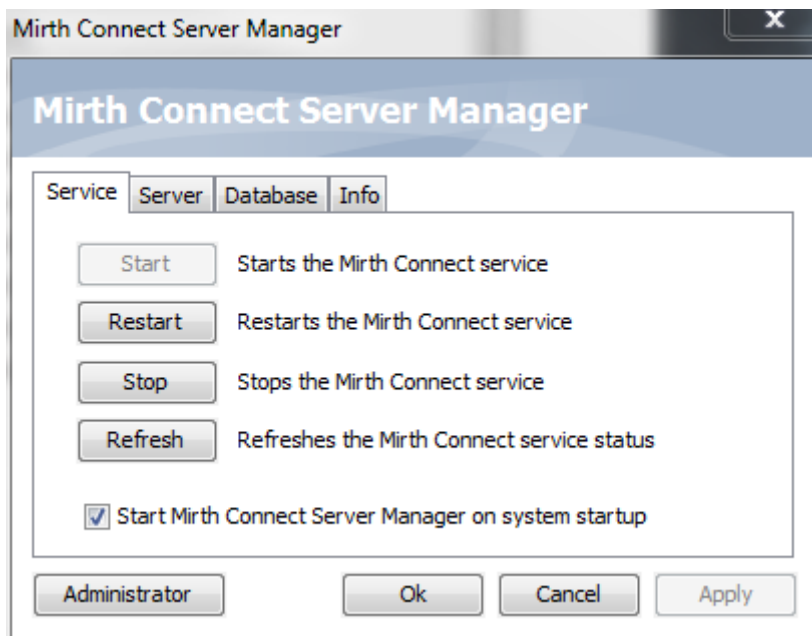


Es wird Ihnen nun die folgende Meldung angezeigt:

**The Mirth Connect Server and Administrator must be restarted before your changes will take effect.**

**Sie können jedoch neue Extensions bereits jetzt installieren! Sie brauchen dann den Schritt 2 der Installation nicht mehr durchzuführen. Kontrollieren Sie jedoch die installierten Extensions nach dem Restart.**

Beenden Sie nun den Mirth Connect Administrator. Führen Sie im Mirth Connect Server Manager einen „Restart“ durch.



Anschließend starten sie den Mirth Connect Administrator neu.

In der Ansicht „Extensions“ sollte nun die SysTek LDT-Extension nicht mehr auftauchen oder die neue Version erscheinen.

Sollte etwas nicht klappen, dann lesen Sie bitte den Abschnitt „Problemlösungen“.

### **Schritt 2 der Installation:**

In der Ansicht „Extensions“ können Sie nun nach der neusten Extension browsen und diese anschließend installieren. Wählen sie im Browse-Fenster nun die Datei „datatype-Ldt.zip“ aus und drücken Sie anschließend den Button „Install“.

Es wird Ihnen nun die folgende Meldung angezeigt:

The Mirth Connect Server and Administrator must be restarted before your changes will take effect.

Beenden Sie nun den Mirth Connect Administrator. Führen Sie im Mirth Connect Server Manager einen „Restart“ durch. Anschließend starten sie den Mirth Connect Administrator neu.

In der Ansicht „Extentions“ sollte nun die SysTek LDT-Extension wieder auftauchen.

#### Installed Plugins

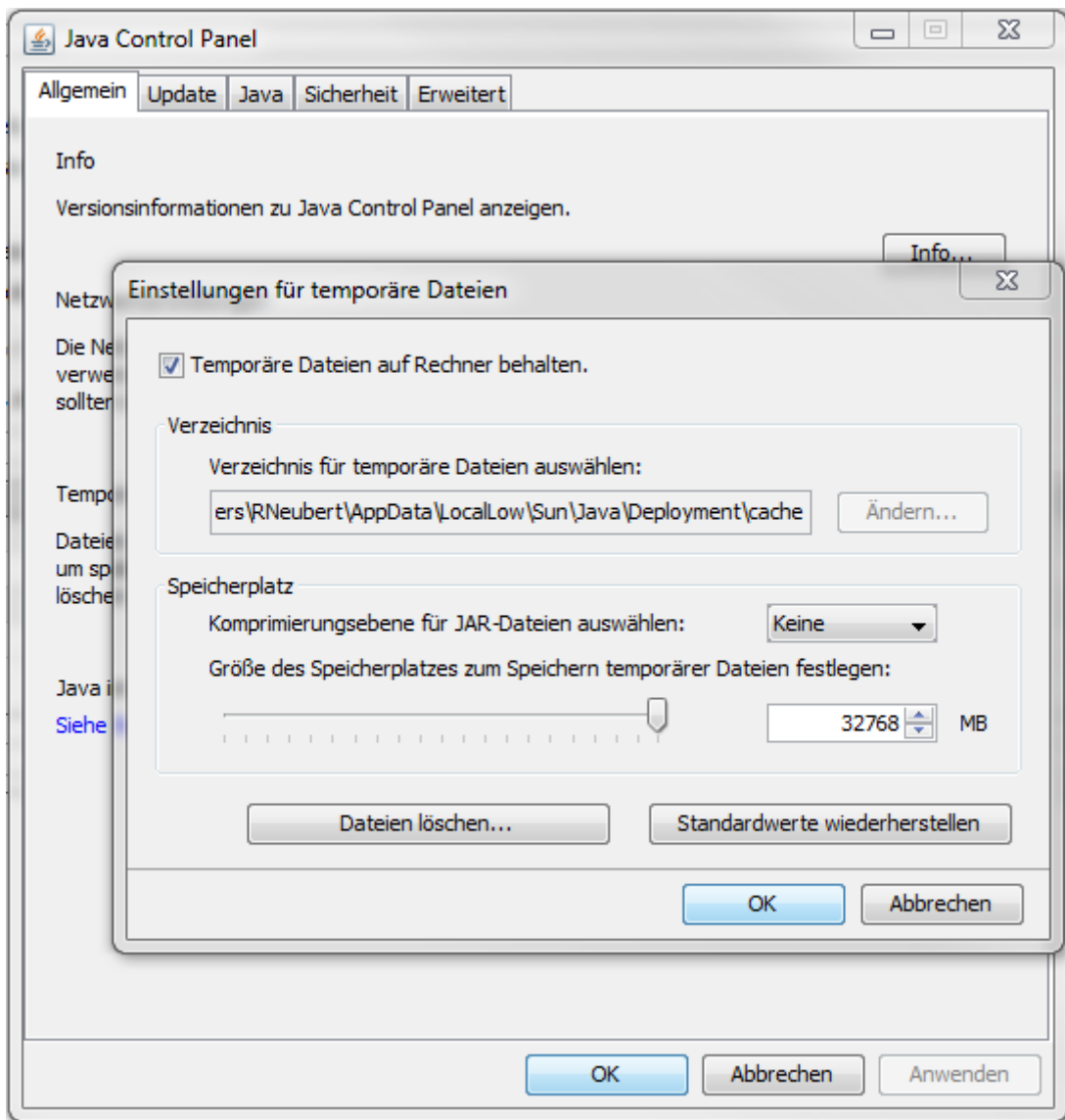
Status	Name	Author ▼	URL	Version	
• Enabled	Message Tree	SysTek GmbH	http://www.systek.de	3.3.1.7856	▲
• Enabled	History	SysTek GmbH	http://www.systek.de	3.3.0.7856	
• Enabled	Report	SysTek GmbH	http://www.systek.de	3.3.0.7801	
• Enabled	Ldt Data Type v2.x	SysTek GmbH	http://www.systek.de	3.3.1.7856	
• Enabled	Ldt Data Type v3.x	SysTek GmbH	http://www.systek.de	3.3.1.7856	
• Enabled	Server Log	Mirth Corporation	http://www.mirthcorp.com	3.3.1.7856	

Alle Schritte sind nun korrekt ausgeführt. Sollte etwas nicht klappen, dann lesen Sie bitte den Abschnitt „Problemlösungen“.

### Problemlösungen für fehlerhafte eine Installation

Beenden Sie den Mirth Connect Server und den Mirth Connect Administrator.

Löschen Sie nun Ihren Java Cache. Rufen Sie dazu „Start“ -> „Java konfigurieren“ auf und wählen Sie „Allgemein“-> „Temporäre Internetdateien“ -> „Einstellungen“:



Wählen Sie „Dateien löschen“.

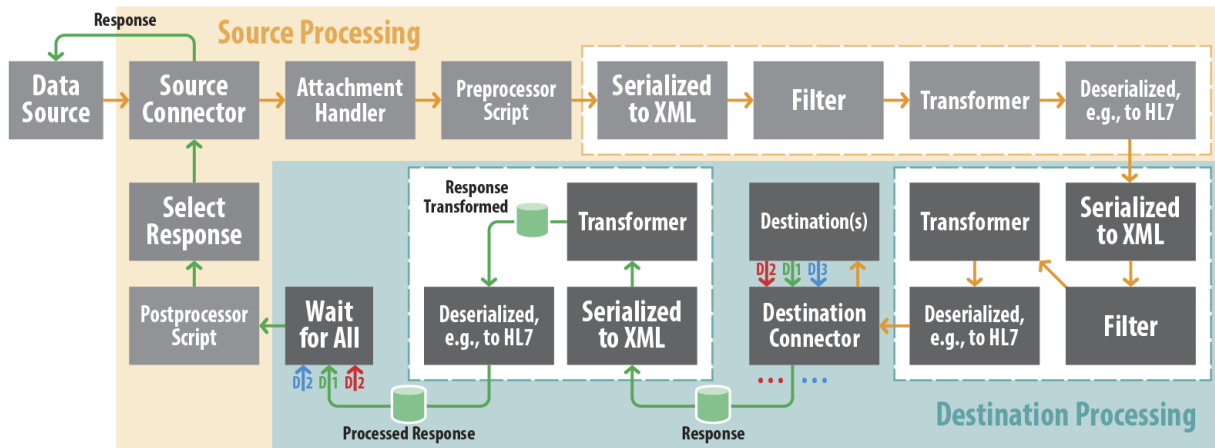
Anschließend starten Sie Ihren Mirth Connect Server und den Mirth Connect Administrator erneut.

In der Regel funktioniert nun alles problemlos.

## **Grundsätzliche Verarbeitung von Nachrichten in Mirth Connect**

Nachrichten durchlaufen in Mirth Connect viele Stationen. Hier sehen Sie eine Workflow-Übersicht:





## Nutzung der SysTek LDT v2.x-Extension

Der LDT v2.x-Datentype kann nun in Ihren Kanälen verwendet werden. Legen Sie einen neuen Kanal an und wählen Sie als „Set Data Types“ „LDT v2.x“ für Ihre Source oder ihre Destination.

**Edit Channel - LDT to LDT**

Summary | Source | Destinations | Scripts

**Channel Properties**

Name: LDT to LDT  Enabled

Data Types:   Clear global channel map on deploy

Dependencies:

**Set Data Types**

Single Edit  Bulk Edit [Expand All](#) | [Collapse All](#)

Connector	Inbound	Outbound
Source Connector	Ldt	Ldt
LDT Empfänger	Ldt	Ldt

**Inbound Properties** (Ldt)

**Serialization**

- Ignore Empty Lines
- Use Strict Parser
- Validate In Strict Parser
- Strip Name Spaces
- Check Byte Length File
- Check Byte Length Record
- Check Byte Length Field

**Batch**

Split Batch By: Record Sets

**Outbound Properties** (Ldt)

**Deserialization**

- Use Strict Parser
- Validate In Strict Parser
- Line delimiter: rn

**Template Serialization**

- Ignore Empty Lines
- Use Strict Parser
- Validate In Strict Parser
- Strip Name Spaces
- Check Byte Length File
- Check Byte Length Record
- Check Byte Length Field

## Inbound Properties

Das „Eingehende“ in einer „Source“ oder „Destination“ ist definiert als „Inbound“. Alle diese Einstellungen beziehen sich nur auf die eingehende Nachricht. So kann zum Lesen der Nachricht, eine andere Einstellung als zum Schreiben der Nachricht gewählt werden. Fehlerhafte Nachrichten können so einfach in fehlerfreie Nachrichten umgewandelt werden.

## Serialization

Dieser Vorgang ist die Transformation von LDT v2.x in ein von SysTek definiertes LDT-XML Format.

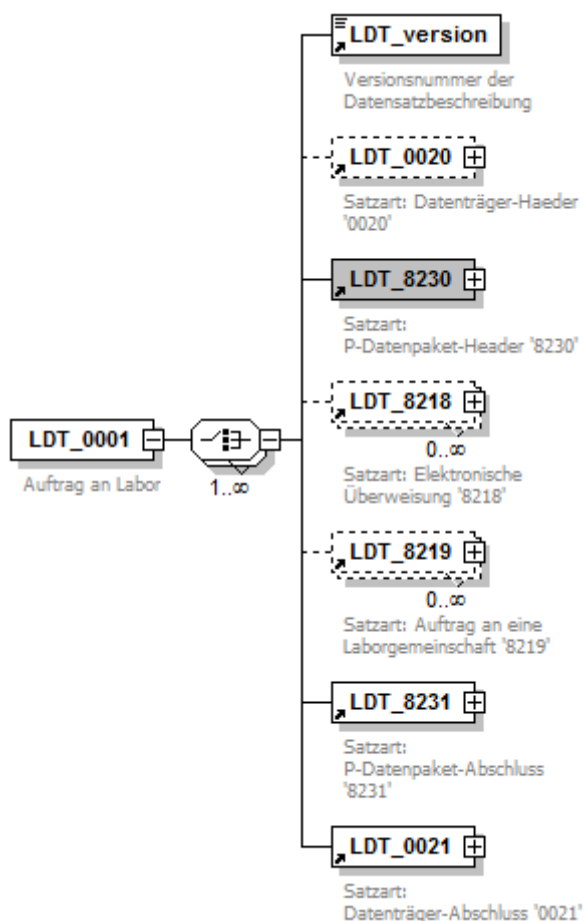
## Ignore empty Lines

Leere Zeilen in der Nachricht werden ignoriert oder verursachen einen Fehler. Bei einem Fehler wird die Nachricht nicht mehr versendet.

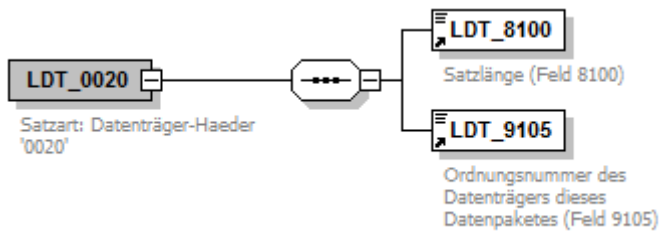
## Use Strict Parser

Ist diese Option aktiviert, werden alle Nachrichten auf die korrekte Position Ihrer Felder geprüft. Als Prüfgrundlage dient die Dokumentation der KBV. SysTek hat diese Dokumentation in XSD Schemas umgesetzt.

Hier das Beispiel eines Auftrages (Auszug):



Jede Satzart ist ebenfalls genau definiert (Auszug).



Bei einem Verstoß im Modus „Strict Parser“ wird die Nachricht nicht mehr versendet. Diese wird bereits „Inbound“ geblockt.

### Validate in Strict Parser

Ist diese Option aktiviert, dann werden alle Feldinhalte und die KBV Regeln geprüft. Dafür sind alle Regeln in XML Dateien definiert:

LDTRules						
xmlns:xsi		http://www.w3.org/2001/XMLSchema-instance				
xsi:noNamespace...		LDTDefinition.xsd				
LDTRule (75)						
ID	Category	Inspection	Inspectionstate	Definition		
1 005	Format	HHMM	F	„Um-Uhrzeit“ HH = Stunde, MM = Minute, Wertebereich: 0000-2359		
2 017	Format	nnnnn	F			
3 021	Format	JJJJMMTT	F	TT= Tag, MM= Monat, JJJJ= Jahr. Zusätzlich erlaubter Wertebereich: JJJJMM00, JJJJ0000, 00000000		
4 024	Format	WWT	F	WW = Wochen, T = Tag, Wertebereich T: 0-6		

Auch alle erlaubten Inhalte sind XML Dateien definiert:

LDT_3108				AllowedContents	
ID	Content	Definition	Content	Definition	
18	LDT_3108	Versichertenart	1	n	3
			1	Mitglied	
			2	Familienversicherter	
			3	Rentner	

Bei einem Verstoß im Modus „Validate Strict Parser“ wird die Nachricht nicht mehr versendet. Diese wird bereits „Inbound“ geblockt.

### Strip Name Spaces

Das Feld mit der ID „99xx“ kann an jeder Position auftauchen. Dafür wurde von SysTek im Schema ein Namespace „unknown“ benutzt. Ohne diesen Namespace könnte die Position des Feldes „99xx“ nicht beliebig sein.

```

<LDT_8218_8410>
  <LDT_8410>GOT</LDT_8410>
  <LDT_8411>Langer Text von GOT</LDT_8411>
  <unknown:LDT_9901>*Information zu GOT 1</unknown:LDT_9901>
  <unknown:LDT_9901>*Information zu GOT 2</unknown:LDT_9901>
</LDT_8218_8410>
<unknown:LDT_9901>*Information for 8218 1</unknown:LDT_9901>
<unknown:LDT_9901>*Information for 8218 2</unknown:LDT_9901>
</LDT_8218>
<LDT_8231>
  <LDT_8100>00044</LDT_8100>
  <LDT_9202>00001087</LDT_9202>

```

```

<unknown:LDT_9901>*Information for 8231 1</unknown:LDT_9901>
<unknown:LDT_9901>*Information for 8231 2</unknown:LDT_9901>
</LDT_8231>

```

Diesen Namespace können Sie mit dieser Option aus der transformierten XML Nachricht entfernen lassen.

### Check Byte Length File

Der Eintrag der Dateilänge wird überprüft. Bei einem Fehler wird die Nachricht nicht mehr versendet.

### Check Byte Length Record

Der Eintrag der Satzlänge wird überprüft. Bei einem Fehler wird die Nachricht nicht mehr versendet.

### Check Byte Length Field

Der Eintrag der Feldlänge wird überprüft. Bei einem Fehler wird die Nachricht nicht mehr versendet.

### Batch

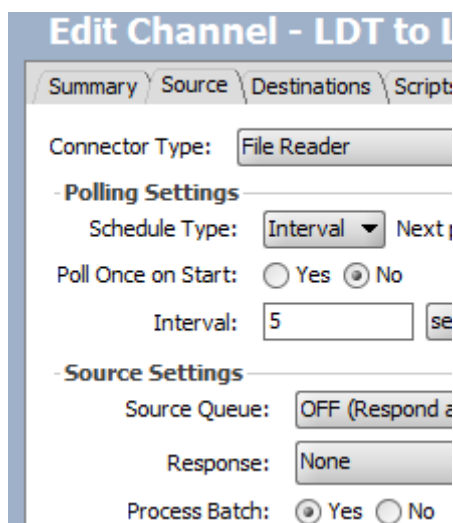
Batch Dateien sind Dateien die mehr als eine Nachricht enthalten. Diese können mit dieser Funktion automatisch in mehreren Nachrichten zerlegt werden.

Hier ein Beispiel einer zusammen kopierten Datei:

ID	LDTRecord								
1	LDTRecord (3) <table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>R8230</td> </tr> <tr> <td>2</td> <td>R8215</td> </tr> <tr> <td>3</td> <td>R8231</td> </tr> </tbody> </table>	ID	Name	1	R8230	2	R8215	3	R8231
ID	Name								
1	R8230								
2	R8215								
3	R8231								
2	LDTRecord (3) <table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>R8230</td> </tr> <tr> <td>2</td> <td>R8215</td> </tr> <tr> <td>3</td> <td>R8231</td> </tr> </tbody> </table>	ID	Name	1	R8230	2	R8215	3	R8231
ID	Name								
1	R8230								
2	R8215								
3	R8231								

Diese Datei enthält 2 Aufträge und kann so in 2 einzelne Nachrichten zerlegt werden.

In der Source TAB muss dafür „Process Batch“ aktiviert sein:



Und eine „Split Batch By“ Option muss in „Set Data Types“ „Inbound“ gewählt sein:

Batch	
Split Batch By	Record Sets

### ***Split Batch by***

Hier bestimmen Sie wie die Datei in einzelne Nachrichten aufgeteilt wird.

### ***Record Sets***

In diesem Fall wird die Datei nun in einzelne Aufträge oder Befunde aufgeteilt. Sind 5 Aufträge oder Befunde in der Datei enthalten, dann werden jeweils 5 Nachrichten produziert. Es handelt sich in der Regel um eine Datei die aus mehreren LDT-Dateien besteht. Diese Datei wurde durch ein „Anhängen“ erzeugt und ist im eigentlichen Sinne eine Batch Datei. Mehre für sich alleine gültigen Nachrichten wurden zu einer Datei zusammenfügt. Jede Nachricht darin ist eventuell valide. Die Datei im gesamten jedoch nicht. Diese kann so aber verarbeitet werden und wird in einzelne valide Nachrichten aufgeteilt.

### ***Tags***

Sind in dieser Datei 5 Befunde für einen Einsender in einer Nachricht zusammenfasst, dann können sie diese Nachricht für diesen Einsender in 5 einzelne Nachrichten zerlegen. So ist immer nur ein Patient in einer Nachricht. Die umgebenden Records der Datei werden vom Plug-In kopiert. Ein „Record Sets Split“ wird vorher durchgeführt. So können aus eine Datei 5 Nachrichten durch „Record Sets Split“ entstehen und jeweils 5 Nachrichten durch „Tag Split“. Eine Datei erzeugt in diesem Fall 25 einzelne Nachrichten.

### ***JavaScript***

Diese Option hat aktuell noch keine Wirkung.

## **Outbound Properties**

Das „Ausgehende“ in einer „Source“ oder „Destination“ ist definiert als „Outbound“. Alle diese Einstellungen beziehen sich nur auf die ausgehende Nachricht. So kann zum Lesen der Nachricht, eine andere Einstellung als zum Schreiben der Nachricht gewählt werden. Fehlerhafte Nachrichten können so einfach in fehlerfreie Nachrichten umgewandelt werden.

### ***Deserialization***

Diese Funktion ist analog zum Inbound Property.

### ***Use Strict Parser***

Diese Funktion ist analog zum Inbound Property.

### ***Validate in Strict Parser***

Diese Funktion ist analog zum Inbound Property.

### ***Line Delimiter***

„Inbound“ werden die „Line delimiter“ automatisch ermittelt. „Outbound“ können sie wählen zwischen „CR+LF“, „CR“ oder „LF“. Die Checksummen werden automatisch neu kalkuliert.

## Template Serialization

Sie haben in Mirth Connect immer die Möglichkeit ein „Inbound Message Template“ und oder ein „Outbound Message Template“ einzusetzen. So können Sie einfach mit der Maus eine Quellnachricht auf eine Zielnachricht transformieren. Inbound könnte das ein LDT v.25 sein und Outbound LDT v2.7 Version LDT1014.01. Templates werden etwas später beschrieben.

### *Ignore empty Lines*

Diese Funktion ist analog zum Inbound Property.

### *Use Strict Parser*

Diese Funktion ist analog zum Inbound Property.

### *Validate in Strict Parser*

Diese Funktion ist analog zum Inbound Property.

### *Strip Name Spaces*

Diese Funktion ist analog zum Inbound Property.

### *Check Byte Length File*

Der Eintrag der Dateilänge wird überprüft. Bei einem Fehler wird die Nachricht nicht als Template akzeptiert.

### *Check Byte Length Record*

Der Eintrag der Satzlänge wird überprüft. Bei einem Fehler wird die Nachricht nicht als Template akzeptiert.

### *Check Byte Length Field*

Der Eintrag der Feldlänge wird überprüft. Bei einem Fehler wird die Nachricht nicht als Template akzeptiert.

## Filter, Transformation, Mappings und Templates

### Templates

Templates sind ein Abbild ihrer Original-Nachrichten die Sie empfangen und versenden möchten. Sie benötigen diese Templates nur wenn die Nachricht zwischen Quelle und Ziel mit der Maus verändert werden muss. Sie können ein Inbound und Outbound Template in jeden Kanal einfügen. Mit dieser Möglichkeit können Sie Nachrichten filtern und verändern. Im Prinzip hilft Ihnen das Template um auf bestimmte Daten in Nachrichten zugreifen zu können. Es stellt so etwas wie ein Muster dar.

Bedenken sie jedoch, dass immer das Outbound Template auch gesendet wird. Alle Felder die Sie nicht überschreiben mit der eingehenden Nachricht werden so gesendet wie es in Ihrem Outbound Template steht. Sie sollten also wenn möglich immer Outbound Templates verwenden die keine wirklichen Daten enthalten.

Ebenso bedenken Sie bitte, dass die eingehende Nachricht im Objekt „msg“ enthalten ist und auch manipulierbar ist. Haben Sie jedoch ein Outbound Template müssen Sie die neuen Daten in das Objekt „tmp“ transformieren.

```
msg['R8220']['F9212'] = „xyz“ (Ohne outbound Template wird xyz gesendet)
tmp['R8220']['F9212'] = „xxx“ (Mit outbound Template wird xxx gesendet)
```

Um Daten Inbound zu Outbound mit Outbound Template zu transformieren müssen Sie dieses Konstrukt nutzen:

```
tmp['R8220']['F9212'] = msg['R8220']['F9212']
```

So überschreiben Sie die Daten Ihres Outbound Templates und die Original Daten des Templates werden geändert.

#	Name	Type
0	L-Datenpaket-Header - verwendeter Zeichensatz (F9106) (out) <-- L-Datenpaket-Header - verwendeter Zeichensatz (F9106) (in)	Message Builder

Step	Generated Script
Message Segment:	tmp[R8220][F9106]
Mapping:	msg[R8220][F9106].toString()

Nach getaner Arbeit können Sie die Templates auch problemlos wieder entfernen.

Hier ein Beispiel als LDT v2.x Format (**Message Templates**):

Reference | Message Trees | **Message Templates**

**Inbound Message Template**

Data Type: Ldt Properties

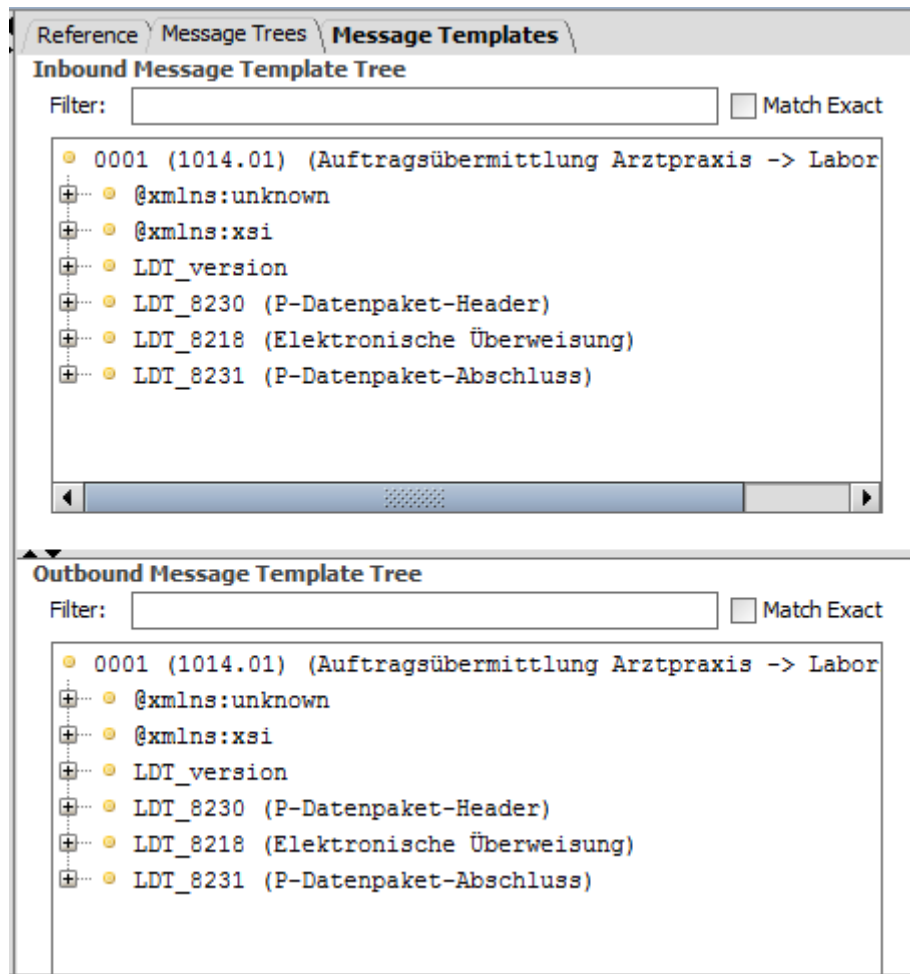
```
01380008230
014810000556
0199212LDT1014.01
0180201251993500
0260203Dialyse Meiderich
0180212888562729
0310211DRMED Christel Lüttger
0180212775103329
0250211 Dirk Hondelmann
0180212764899329
```

**Outbound Message Template**

Data Type: Ldt Properties

```
01380008230
014810000556
0199212LDT1014.01
0180201251993500
0260203Dialyse Meiderich
0180212888562729
0310211DRMED Christel Lüttger
0180212775103329
0250211 Dirk Hondelmann
0180212764899329
```

Mirth Connect wandelt das Original Format der LDT v2.x Nachrichten nun in ein SysTek basiertes XML Format um. Dieses sieht dann wie folgt aus (**Message Trees**):

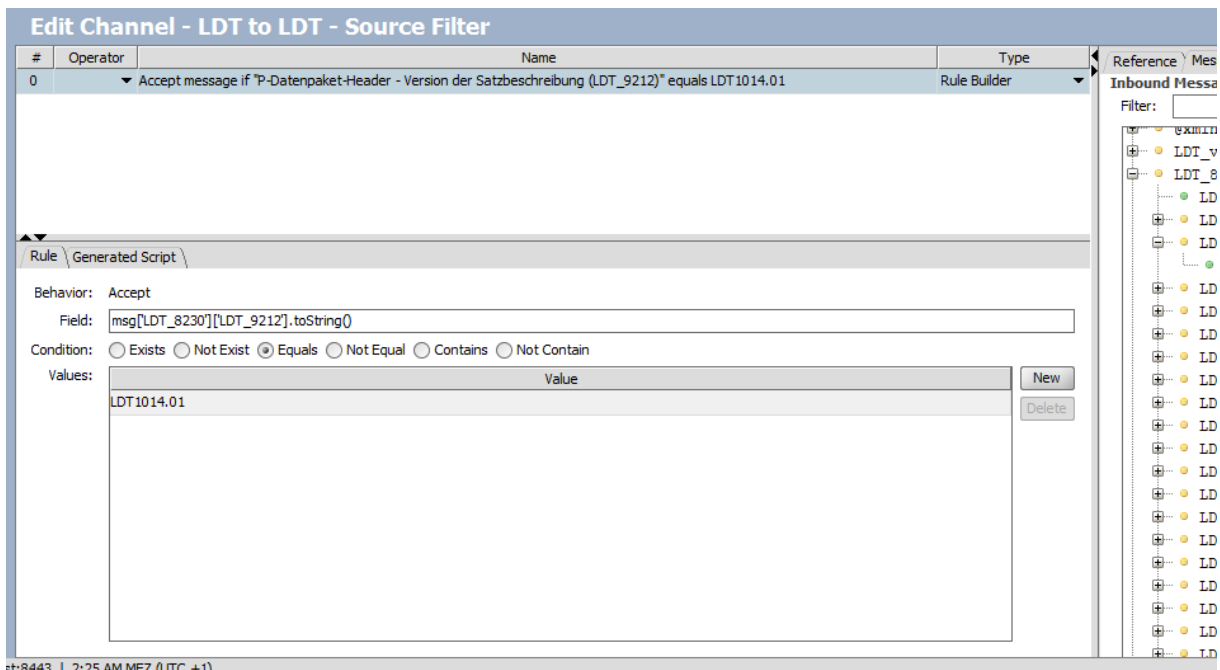


In diesem Baum können Sie nun sehr einfach navigieren und Felder mit der Maus im Drag-und-Drop Verfahren verbinden oder Filter erstellen. Wir haben uns größte Mühe gegeben ihnen die Nachricht sinnvoll und intuitiv zu präsentieren.

## Filter

Ein Filter kann Nachrichten von der Quelle oder auch an bestimmte Ziele ausfiltern. Beispielhaft wird hier ein Filter angezeigt, der Nachrichten mit einer zu altem LDT Versionen ausfiltert (Nur Nachrichten mit Version „LDT1014.01“ werden akzeptiert).

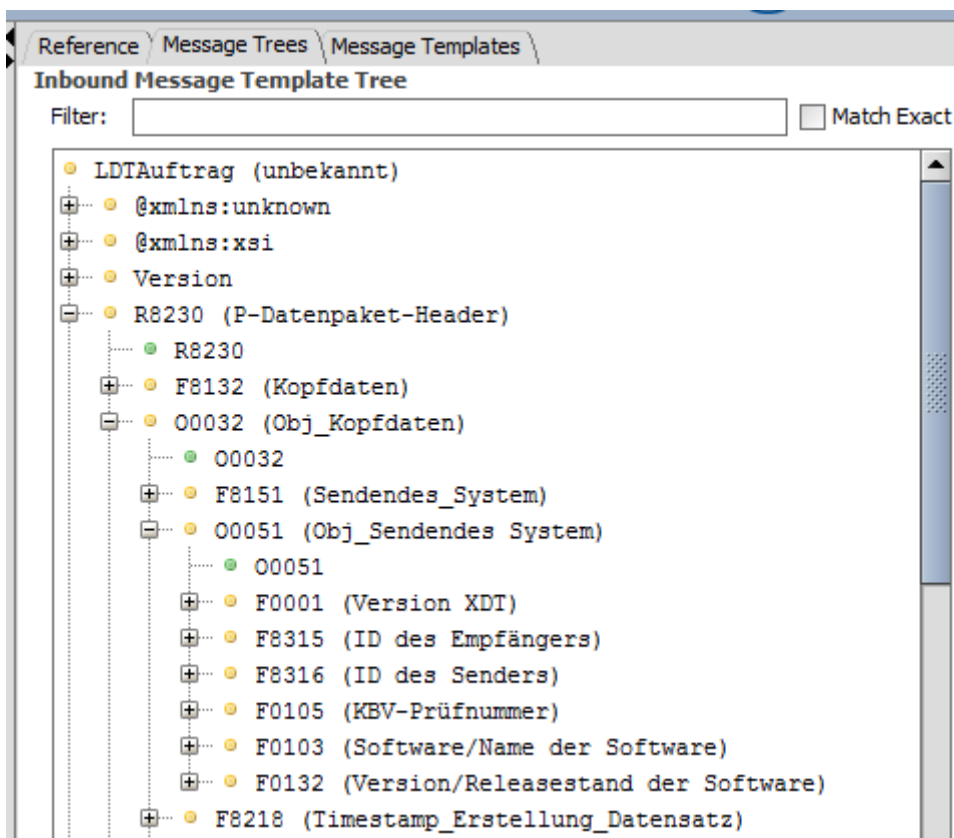




Jede Nachricht die nicht diesem Format entspricht wird ausgefiltert. Die Komplexität von Filtern lässt sich für alle Belange beliebig erweitern. Es können beliebig viele Filter angelegt werden. Diese Filter können bereits in der Quelle greifen oder erst an einem beliebigen Ziel. So können Sie einem Ziel die alten Nachrichten zuführen und ein anders Ziel vor alten Nachrichten schützen.

## SysTek Extension „LDT v3.x“

Vollständige Adaption von LDT v3.x zu XML



## SysTek Extension „Report“

Vollständige Dokumentation Ihres Kanals.

### Channel Summary

#### Channel properties

Channel properties	
id	6029d68b-c215-431c-b0ff-9b4bda6ddcfb
nextMetaDataId	3
name	LDT2 over XML to LDT2
enabled	true
revision	15

#### Source Overview

Name	Connector Type	Inbound Type	Outbound Type	Filter Rules	Transformer Steps
sourceConnector	File Reader	LDT v2.x	XML	0	0

#### Destinations Overview

Name	Connector Type	Inbound Type	Outbound Type	Filter Rules	Transformer Steps
LDT v2.x Filewriter	File Writer	XML	LDT v2.x	0	0
LDT XML v2.x Filewriter	File Writer	XML	XML	0	0

#### Source Connector

##### 2.0 File Reader

## SysTek Extension „History“

Vollständige Historie Ihrer Arbeit an einem Kanal und die Möglichkeit Ihre Arbeit zu vergleichen und rückgängig zu machen.

The screenshot shows a 'Channel History' window with two main panes. The left pane contains a table of revisions, and the right pane shows the XML configuration for the selected revision.

Diff	Rev	Date & Time	User	
✓	168	2016-02-19 17:53:27.0	admin	<...>
✓	170	2016-02-19 18:59:07.0	admin	<...>
✓	171	2016-02-19 21:27:23.0	admin	<...>
✓	172	2016-02-19 21:36:36.0	admin	<...>
✓	186	2016-02-20 02:20:40.0	admin	<...>
✓	196	2016-02-20 03:57:19.0	admin	<...>
✓	391	2016-03-10 18:36:01.0	admin	<...>
✓	401	2016-03-15 18:23:41.0	admin	<...>
✓	416	2016-03-16 01:18:58.0	admin	<...>
✓	417	2016-03-16 01:21:11.0	admin	<...>
✓	423	2016-03-18 00:43:26.0	admin	<...>
✓	431	2016-03-21 04:05:33.0	admin	<...>
✓	440	2016-03-22 19:53:48.0	admin	<...>
✓	441	2016-03-23 19:15:34.0	admin	<...>

```

<channel version="3.3.1">
  <id>6029d68b-c215-431c-b0ff-9b4bda6ddcfb</id>
  <nextMetadataId>3</nextMetadataId>
  <name>LDT2 over XML to LDT2</name>
  <description></description>
  <enabled>true</enabled>
  <lastModified>
    <time>1458756934725</time>
    <timezone>Europe/Berlin</timezone>
  </lastModified>
  <revision>15</revision>
  <sourceConnector version="3.3.1">
    <metaDataId>0</metaDataId>
    <name>sourceConnector</name>
    <properties class="com.mirth.connect.connectors.file.FileReceiverProperties" version="3.3.1">
      <pluginProperties/>
      <pollConnectorProperties version="3.3.1">
        <pollingType>INTERVAL</pollingType>
        <pollOnStart>false</pollOnStart>
        <pollingFrequency>5000</pollingFrequency>
        <pollingHour>0</pollingHour>
        <pollingMinute>0</pollingMinute>
      </pollConnectorProperties>
      <pollConnectorPropertiesAdvanced>
        <weekly>true</weekly>
        <inactiveDays>
          <boolean>>false</boolean>
          <boolean>>false</boolean>
          <boolean>>false</boolean>
          <boolean>>false</boolean>
          <boolean>>false</boolean>
          <boolean>>false</boolean>
          <boolean>>false</boolean>
          <boolean>>false</boolean>
          <boolean>>false</boolean>
        </inactiveDays>
        <dayOfMonth>1</dayOfMonth>
        <allDay>true</allDay>
        <startingHour>8</startingHour>
        <startingMinute>0</startingMinute>
        <endingHour>17</endingHour>
        <endingMinute>0</endingMinute>
      </pollConnectorPropertiesAdvanced>
    </pollConnectorProperties>
    <sourceConnectorProperties version="3.3.1">
      <responseVariable>None</responseVariable>
      <respondAfterProcessing>true</respondAfterProcessing>
      <processBatch>true</processBatch>
      <firstResponse>false</firstResponse>
      <resourceIds class="singleton-set">
        <string>Default Resource</string>
      </resourceIds>
    </sourceConnectorProperties>
  </sourceConnector>
  <scheme>FILE</scheme>
</channel>

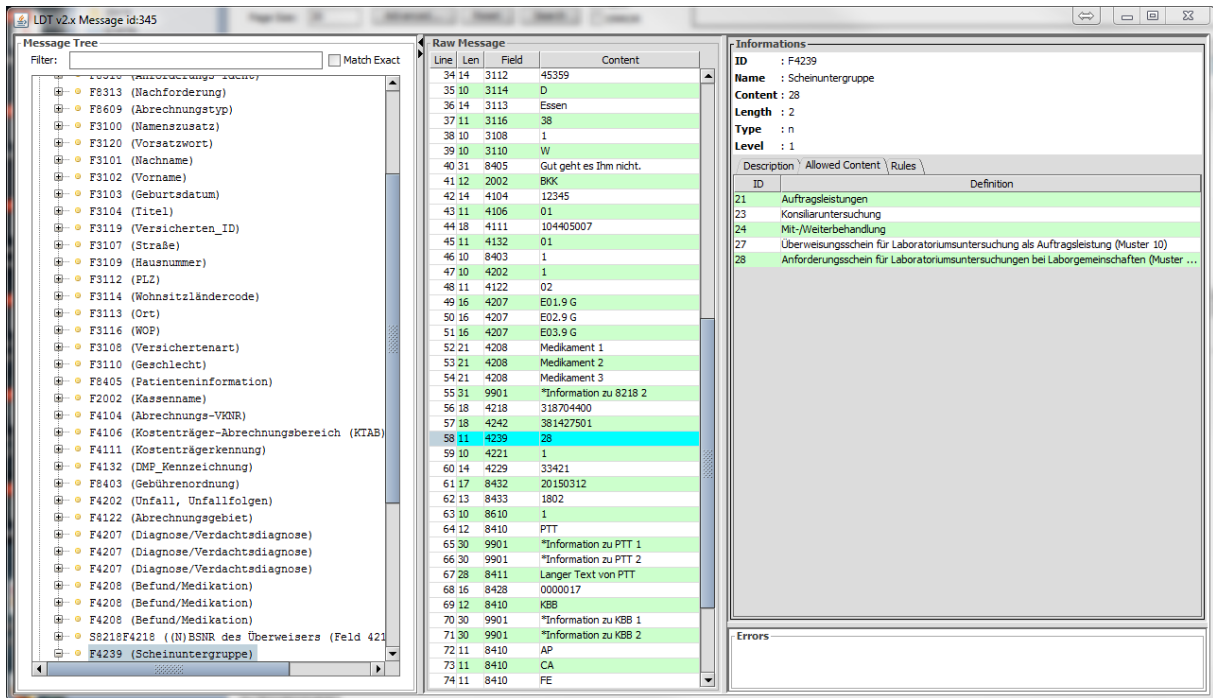
```

At the bottom of the window, there are buttons for 'Prune', 'Compare', 'Revert', and 'Close'. The status bar indicates '14 revisions'.

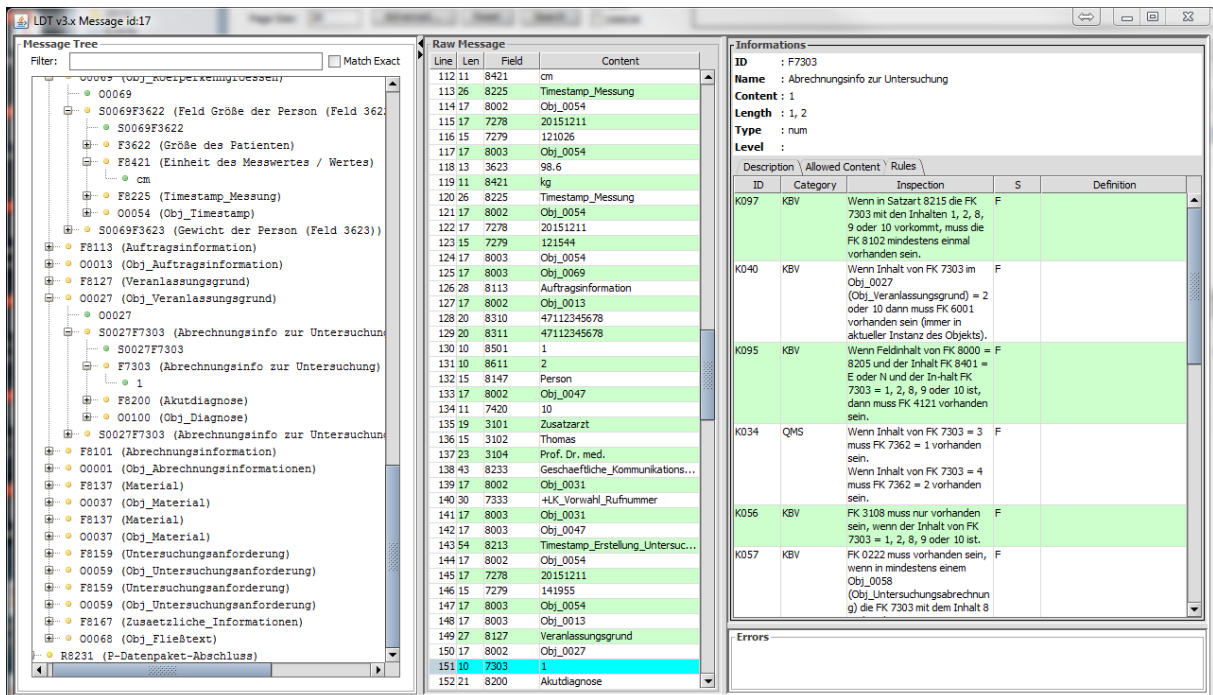
## SysTek Extension „Message Tree“

Diese Extension zeigt Ihnen jede LDT Nachricht KBV konform an. Sie können alle Regeln und erlaubten Inhalte sehen. Sie können in der Tree oder Raw Ansicht navigieren. Fehler werden Ihnen ebenfalls angezeigt.

LDT Ansicht von jeder LDT v2.x Nachricht:



LDT Ansicht von jeder LDT v3.x Nachricht:



## Schlusswort

Fast alles ist mit dieser Architektur möglich. Die Möglichkeiten, den passenden Workflow zu entwickeln, sind vielfältig und über viele Wege realisierbar. SysTek verfügt über das Know-how, den für Sie effektivsten Workflow zu adaptieren. SysTek verfügt über eine umfassende Wissens- und Installationsbasis in allen Bereichen der Labor-EDV.

Gerne unterbreiten wir Ihnen ein Konzept für Ihre Kommunikationslandschaft.

Gerne schulen wir Sie anschließend, damit Sie die meisten Kanäle selbst erstellen können.

**SysTek ist offizieller „referral Partner“ der Firma mirth corporation (USA).**

**SysTek ist offiziell von der Firma mirth corporation (USA) zertifiziert (MC00269).**

Mirth Connect™ ist ein Produkt der Firma mirth corporation (USA).