einheitliches XML-basiertes Transportverfahren

Neue eXTra Standardnachrichten

RepeatResponse
StatusRequest
StatusResponse
ListOfStatusResponse

Version 1.0
Final

Herausgeber:

AWV – Arbeitsgemeinschaft für wirtschaftliche Verwaltung e. V.
Düsseldorfer Str. 40
65760 Eschborn
Vereinsregister 73 VR 5158, Amtsgericht Frankfurt am Main
Telefon: 0 61 96/7 77 26-0
Fax: 0 61 96/7 77 26-51
Mail: info@awv-net.de
Web: www.extra-standard.de, www.awv-net.de.

Das vorliegende Dokument „Neue eXTra Standardnachrichten“ zum einheitlichen XML-basierten Transportverfahren „eXTra“ wurde von Mitarbeiterinnen und Mitarbeitern des AWV-Arbeitskreises 2.1 „Vereinheitlichung von Datenübermittlungssystemen“ im Fachausschuss 2 „Verwaltungsvereinfachung und Entbürokratisierung im personalwirtschaftlichen Umfeld“ entwickelt.

Inhalt

1. Motivation

Die Erfahrungen im realen Betrieb mit dem eXTra-Standard sowie Überlegungen zur Nutzung des eXTra-Standards für neue Verfahren mit erweiterten Anforderungen zeigen Szenarien auf, die eine Erweiterung des eXTra-Standards nahelegen. Dies wird im Folgenden anhand verschiedener Szenarien erläutert.

Szenario 1

Im aktuellen eXTra-Standard der Version 1.2 kann es vorkommen, dass für den Sender nach einem eXTra-Request ein für ihn unbekannter Zustand beim Empfänger eintritt und der Sender keine eXTra-spezifischen Mittel hat, um diesen Zustand in Erfahrung zu bringen.

Folgendes Szenario 1.1 sei gegeben:

Der Sender hat eine fachliche Nachricht mit einem eXTra-Request der Art scenario=request-with-acknowledgement abgesendet und als Rückantwort keine eXTra-Response, sondern eine Fehlermeldung der DFÜ-Ebene, z.B. einen Timeout der http(s)-Instanz, erhalten. Für den Sender ist jetzt nicht entscheidbar, ob der Empfänger die vollständige Nachricht nicht empfangen und ablegen konnte, der Sender die Nachricht also noch einmal senden kann. Oder ob der Empfänger die Nachricht zwar empfangen und ablegen konnte, aber die eXTra-Response aus welchem Grund auch immer nicht zurückgeben konnte, z.B. weil der Proxy oder die DFÜ-Ebene auf einen Timeout lief. Wenn nun der Sender auf Grund der Fehlermeldung die Nachricht nochmals sendet, hat der Empfänger die Nachricht doppelt erhalten bzw. das Fachverfahren (z.B. das DEÜV-Verfahren) lehnt später die Nachricht wegen Doppellieferung ab (beim DEÜV-Verfahren wegen einer falschen laufenden Dateinummer).

Wenn der Sender ein eXTra-spezifisches Sprachmittel hätte, um vom Empfänger exakt die damalige Antwort auf seinen eXTra-Request mit scenario=request-with-acknowledgement anfordern und erneut erhalten zu können, dann könnte er auch auf die Fehlermeldung sofort automatisch, also ohne menschliche Interaktion (z.B. Anruf bei der Hotline) gezielt und korrekt reagieren. Ziel ist es deshalb, ein geeignetes eXTra-spezifisches Sprachmittel zur Verfügung zu stellen.

Folgendes Szenario 1.2 sei gegeben:

Der Sender hat eine fachliche Nachricht mit einem eXTra-Request der Art scenario=request-with-response abgesendet und als Rückantwort keine eXTra-Response, sondern eine Fehlermeldung der DFÜ-Ebene, z.B. einen Timeout der http(s)-Instanz erhalten. Für den Sender ist jetzt nicht entscheidbar, ob das Fachverfahren beim Empfänger die vollständige Nachricht nicht erhalten und verarbeiten konnte, der Sender die Nachricht also noch einmal senden kann, um die gewünschte Antwort zu erhalten. Oder ob das Fachverfahren beim Empfänger die Nachricht zwar erhalten und verarbeiten konnte, aber die eXTra-Response aus welchem Grund auch immer nicht zurückgeben konnte, z.B. weil der Proxy oder die DFÜ-Ebene auf einen Timeout lief.

Die Frage ist nun, wie die Unsicherheit des Senders zu bewerten ist. Entscheidend hierfür ist, welche Bedeutung der eXTra-Request mit scenario=request-with-response hatte. Wurde mit diesem eXTra-Request eine Auskunftsfunktion des Fachverfahrens auf Empfängerseite angesprochen – also eine Funktion, die den Systemzustand des Fachverfahrens auf Empfängerseite nicht verändert – so kann der Sender den ursprünglichen eXTra-Request einfach wiederholen; dafür ist kein neues eXTra-Sprachmittel erforderlich.

Soll jedoch mit diesem eXTra-Request eine Veränderung des Systemzustands des Fachverfahrens auf Empfängerseite durch eine Verarbeitung der übermittelten fachlichen Daten bewirkt werden, so kann es sein, dass – je nach Ausgestaltung des Fachverfahrens auf Empfängerseite – der Sender auf die nicht erhaltene Response nicht verzichten kann, wenn er weiterhin in jedem Fall korrekte eXTra-Requests senden will bzw. wenn eine Wiederholung des ursprünglichen eXTra-Requests zu einem fehlerhaften Systemzustand des Fachverfahrens auf Empfängerseite führen wurde. In einem derartigen Fall benötigt der Sender - analog zum obigen Szenario 1.1 - ein eXTra-spezifisches Sprachmittel, um vom Empfänger exakt die damalige Antwort auf seinen eXTra-Request mit scenario=request-with-response anfordern und erneut erhalten zu können.

Aufgrund der Tatsache, dass das Szenario 1.2 bei Verwendung der aktuell registrierten verbundspezifischen eXTra-Standards keine Relevanz hat – ein eXTra-Request mit der eXTra Standardnachricht DataRequest ist als Anforderung einer Auskunft einzustufen, die keine Veränderung des Systemzustands bewirkt (bis zu dem Zeitpunkt, an dem eine ConfirmationOfReceipt erfolgt) – wird im Folgenden lediglich das Szenario 1.1 genauer betrachtet.

Szenario 2

Der Sender hat im aktuellen eXTra-Standard der Version 1.2 keine Möglichkeit sich über den aktuellen Zustand oder Bearbeitungsstand seiner fachlichen Nachrichten zu erkundigen. Derzeit fehlen eXTra-spezifische Sprachmittel für Recherchen und für den Nachvollzug von Vorgängen der Vergangenheit. Fragestellungen in der Art „was ist mit meiner Sendung mit der RequestID/ResponseID =xyz vor 3 Wochen passiert, für welche die Verarbeitungsrückmeldung überfällig ist“ können derzeit nicht unterstützt werden.

Analoges gilt für eine effektive Unterstützung für den Nachvollzug und die Diagnose insbesondere von ungeklärten bzw. rätselhaften Vorgängen. Bei der obigen Fragestellung überfälliger Verarbeitungsrückmeldungen wäre es für beide Seiten hilfreich, wenn der Sender im Fehlerfall zusätzlich die Stelle, an der die Verarbeitungskette beim Empfänger abbrach, genau fixieren und zudem den Grund des Abbruchs in Erfahrung bringen könnte – ohne dafür auf Empfängerseite langwierig auf die Suche in diversen (Log-) Protokollen gehen zu müssen. Die Motivation ist in diesem Fall eine Reduzierung des Aufwands für den Sender und den Empfänger (bzw. dessen Hotline).
Es geht also um die Klärung von Vorgängen, die nicht aktuell gegeben sind, sondern in der Vergangenheit liegen. Dies umfasst auch die Vorgänge, die bereits unter dem Stichwort „Ackowledgement2“ diskutiert wurden, wobei die nun erfolgende allgemeinere Betrachtung zu einem eleganteren Lösungsansatz führt.

Eine weitere Überlegung führt ebenfalls dazu, die eXTra-Sprachmittel in diesem Sinn zu erweitern. Wenn mit der Übermittlung fachlicher Nachrichten nicht nur die technische Verarbeitung durch das Fachverfahren angestoßen wird, sondern sich darüber hinaus eine evtl. langwierige Begutachtung - z.B. durch einen menschlichen Prüfer - anschließt, wäre es vorteilhaft, wenn es eine Informationsmöglichkeit zum aktuellen Stand der Bearbeitung durch das Fachverfahren (oder den menschlichen Bearbeiter) und eine abschließende Rückmeldung über den Abschluss des gesamten Verfahrens (und dem Löschen der übermittelten Daten beim Empfänger) gäbe.

Weiterhin gibt es noch die Anforderung von Softwarehäusern, den Anwender auch für den Fall zu unterstützen, dass auf Grund eines technischen Defekts oder sonstigen Katastrophe beim Anwender dessen Datenbestände verloren gegangen sind und beim Einspielen der Sicherungsdatenbestände nicht der Zustand zum Zeitpunkt der Katastrophe hergestellt werden

konnte, sondern nur ein Zeitpunkt davor. Sofern sich der Anwender nun über den Stand seiner letzten Sendungen und Bestätigungen informieren könnte, wäre für ihn eine definierte Fortführung der Arbeit deutlich erleichtert.
Auch diese Anforderung führt zu Überlegungen, die eXTra Sprachmittel zu erweitern.

Abgrenzung

Die Idee, für Recherchen und den Nachvollzug die eXTra-Sprachmittel „Logging“ einzusetzen, ist zwar prinzipiell denkbar, jedoch nicht ausreichend, weil es bislang noch keine Sprachmittel für den Sender gibt, in dem er die Logging-Einträge auf Empfängerseite für Vorgänge in der Vergangenheit anfordern könnte. Zudem ist zu bedenken, dass die Logging-Funktion sehr mächtig und deswegen im laufenden Betrieb recht aufwändig ist. Deswegen ist die Wahrscheinlichkeit sehr gering, dass der Sender wie auch der Empfänger für alle Sendevorgänge die Logging-Funktion permanent eingeschaltet lässt – gerade dieses wäre aber notwendig, wenn man Vorgänge der Vergangenheit nachvollziehen will.

2. Entwurf

2.1. Prinzipielle Überlegungen

Ausgestaltung als eXTra-Standardnachrichten

Die eXTra-Sprachmittel, die der Sender beim Szenario 1 für eine Wiederholung der Antwort, bzw. beim Szenario 2 im Sinne einer Auskunftsfunktion benötigt, werden in Form von zwei neuen eXTra-Standardnachrichten ausgestaltet:

- „RepeatResponse“ für Szenario 1 und
- „StatusRequest“ für Szenario 2.

Szenario 1: RepeatResponse, Anforderung zur Wiederholung der Antwort

Die neue Standardnachricht RepeatResponse fordert vom Empfänger die Wiederholung genau der Antwort an, die der Sender für einen vorherigen Request hätte erhalten sollen, aber aufgrund eines Fehlers - z.B. auf der DFÜ-Ebene - nicht erhalten hat. Dieser ursprüngliche Request könnte im Prinzip eine Lieferung im Zuge eines Sendeprozesses sein, er könnte aber auch die konkrete Anforderung eines Holprozesses mittels DataRequest oder die Bestätigung einer vorangegangenen erfolgreichen Abholung mittels ConfirmationOfReceipt sein. Die Notwendigkeit, vom Empfänger eine exakte Wiederholung einer Antwort mittels RepeatResponse anzufordern, besteht nur für den Sendeprozess, nicht jedoch für einen Request mittels DataRequest oder ConfirmationOfReceipt, denn diese Requests kann der Sender ohne jegliche Einschränkung und ohne Gefahr einer Fehlinterpretation einfach wiederholen.

Auch aus dieser Betrachtung heraus wird die neue Standardnachricht RepeatResponse zunächst nur für die Anwendung im Rahmen des Sendeprozesses und nur mit scenario=request-with-acknowledgement betrachtet. Unabhängig davon wird versucht die Behandlung des RepeatResponse so vorzunehmen, dass eine Erweiterung um scenario=request-with-response ermöglicht wird.

Szenario 1.1: Rückmeldung des Empfängers auf ein RepeatResponse, also auf eine Anforderung zur Wiederholung der Antwort

Gefordert ist die exakte Wiederholung der Antwort auf einen Request mit scenario=request-with-acknowledgement oder mit scenario=request-with-response. Die Wiederholung der Antwort muss hierbei im TransportBody der eXTra-Response zurückgegeben werden, um den formalen Ansprüchen einer eXTra-Response zu genügen.

Wenn der ursprüngliche Request des Senders mit scenario=request-with-acknowledgement nur aus der Transportebene bestand, muss die Rückmeldung des Empfängers im TransportBody nur die Response der damaligen Transportebene, also den Response-TransportHeader, enthalten. Allgemeiner formuliert: Abhängig davon, wie viele Ebenen der ursprüngliche Request des Senders umfasst hat (z.B. Transport- und Paketebene), muss die Rückmeldung des Empfängers im TransportBody ebenso viele Ebenen umfassen (z.B. bei der GKV Response-TransportHeader und n Response-PackageHeader).

Szenario 2: Rückmeldung des Empfängers auf ein StatusRequest, also auf eine Anforderung zu einer Recherche

Prinzipiell könnte man es dem Empfänger überlassen, wie er auf die Anforderung eines StatusRequest reagiert und welche Informationen er zurückmeldet. Im Sinne einer Standardisierung von Vorgängen und deren Bearbeitung ist jedoch der Ansatz vorzuziehen, auch für die Rückmeldung eine weitere dritte eXTra-Standardnachricht – Vorschlag für den Namen ist StatusResponse - zu entwickeln.

Der Empfänger meldet als Antwort auf einen StatusRequest den zuletzt erreichten Zustand der betroffenen Sendungen zurück. Je nachdem wie die Empfängerseite ausgestaltet ist, kann der zuletzt erreichte Zustand durch

a) den eXTra Distribution-Server im Minimalausbau,
b) das Fachverfahren oder
c) sogar den eXTra Delivery-Server im Endausbau

gemeldet werden.

Im Fall a) im Minimalausbau, bei dem nur der eXTra Distribution-Server den zuletzt erreichten Zustand melden kann, lautet im fehlerfreien Normalfall die Antwort sinngemäß „eXTra

Distribution-Server: Weitergabe an das Fachverfahren erfolgreich durchgeführt“. Im Fehlerfall könnte die Antwort z.B. „eXTra Distribution-Server: Fehler beim Entschlüsseln – keine Weitergabe an das Fachverfahren“ lauten.

Im Fall b), bei dem das Fachverfahren in die Rückmeldung zu einem StatusRequest integriert ist, lautet im fehlerfreien Normalfall die Antwort sinngemäß z.B. „Fachverfahren: Daten erfolgreich erhalten“ oder „Fachverfahren: Daten erfolgreich verarbeitet“. Im Fehlerfall könnte die Antwort z.B. „Fachverfahren: Fehler beim Verarbeiten der Daten“ lauten.

Im Fall c), bei dem der Weg der gesendeten fachlichen Daten über das Fachverfahren bis hin zum eXTra Delivery-Server verfolgt werden kann, lautet im fehlerfreien Normalfall die Antwort sinngemäß z.B. „eXTra Delivery-Server: Rückmeldung vom Fachverfahren erfolgreich erhalten“ oder „eXTra Delivery-Server: DataRequest erfolgreich ausgeliefert“ oder „eXTra Delivery-Server: ConfirmationOfReceipt erfolgreich erhalten“. Im Fehlerfall könnte die Antwort z.B. lauten: „eXTra Delivery-Server: letzter ConfirmationOfReceipt fehlerhaft“.

2.2. Die neue eXTra-Standardnachricht RepeatResponse

Die neue Standardnachricht **RepeatResponse** kann für die Spezifikation, auf welche Sendung sich die gewünschte zu wiederholende Antwort bezieht, auf die Sprachmittel eines eXTra-Requests (der Transportebene) zurückgreifen.
Analoges gilt auch für die eXTra-Response im TransportBody als Antwort auf einen eXTra-Request mittels **RepeatResponse:** Auch hier stehen bereits alle erforderlichen Sprachmittel aus dem eXTra Basisstandards zur Verfügung.

Laut Definition ist ein eXTra-Request eineindeutig über die Sender- und RequestID (und evtl. dem Timestamp) im TransportHeader identifizierbar, unabhängig von der Anzahl der Ebenen; aus denen der eXTra-Request besteht. Deshalb genügt es auch, wenn beim eXTra-Request mittels **RepeatResponse** lediglich die damalige Transportebene, in der Regel nur der damalige TransportHeader, angegeben wird. Je nach Ausgestaltung des Empfängers kann es sinnvoll sein, neben dem TransportHeader auch die damaligen PlugIns (z.B. das PlugIn DataSource) anzugeben. Weitere (eigene) Sprachmittel benötigt die neue Standardnachricht **RepeatResponse** nicht. Es genügt also, den damaligen eXTraRequest-Aufruf mit den Elemente der TransportEbene unter die Elemente Data und ElementSequence bzw. Base64CharSequence einzuklinken. Dabei ist es ratsam, vom damaligen eXTraRequest-Aufruf auch das Rootelement Transport (empfohlene Schreibweise ab eXTra 1.3) bzw. XMLTransport (bis eXTra 1.2) einzufügen, um damit auch die Konstellation abfangen zu können, bei der zwischen damaligem eXTraRequest-Aufruf und jetzigen RepeatResponse Aufruf ein eXTra Versionswechsel stattfand.

Damit ergibt sich für den eXTra-Request mittels **RepeatResponse** im TransportBody folgende Grobstruktur:

```
<xcpt:Data“>
        <xcpt: Base64CharSequence>
            <xreq:Transport version=”1.3” … >
                <xreq:TransportHeader> ……</xreq:TransportHeader>
                <xreq:TransportPlugIns> ……</xreq:TransportPlugIns>
                 <xreq:TransportBody/>
            <xreq:XMLTransport>
        </xcpt: Base64CharSequence>
</xmsg:Data>
```

Für die eXTra-Response als Antwort auf einen eXTra-Request mittels **RepeatResponse** ergibt sich im TransportBody folgende Grobstruktur (wenn der ursprüngliche eXTra-request z.B. zwei Ebenen - Transport- und Paketebene – umfasste):

```
<xcpt:Data“>
        <xcpt:ElementSequence>
            <xres:Transport version=”1.3” … >
                <xres:TransportHeader> ……</xres:TransportHeader>
                <xres:TransportPlugIns> ……</xres:TransportPlugIns>
                 <xres:TransportBody>
                    <xres:Package>
                                <xres:PackageHeader> ……</xres:PackageHeader>
                                <xres:PackagePlugIns> ……</xres:PackagePlugIns>
                                <xres:PackageBody/>
                    </xres:Package>
                 </xres:TransportBody>
            </xres:XMLTransport>
        </xcpt:ElementSequence>
</xmsg:Data>
```

Hinweis: Eine Besonderheit von RepeatResponse liegt darin, dass die Nachricht keine eigenen Sprachmittel benötigt. Eine Spezifikation von RepeatResponse in der Schemadatei der eXTra Standardnachrichten ist nicht erforderlich, weder für den eXTra-Request noch für den eXTra-Response. Aufgrund dessen bleibt die Designentscheidung des eXTra Standards erhalten, mit der eine strikte Trennung der Schemadateien für den eXTra-Basisstandard von der Schemadatei für die eXTra Standardnachrichten festgelegt wurde.

2.2.1. Beispiel für die neue Standardnachricht RepeatResponse

Im folgenden Beispiel wird auch der Fall gezeigt, dass sich die eXTra Version vom ursprünglichen eXTra-Request bis zur späteren Anforderung mittels RepeatResponse von 1.1 auf 1.3 geändert hat.

Annahme: Der ursprüngliche eXTra-Request hatte folgende Gestalt:

```
<xreq:XMLTransport version=”1.1” …>
        <xreq:TransportHeader>
            <xcpt:TestIndicator> …</xcpt:TestIndicator>
            <xcpt:Sender> …</xcpt:Sender>
            <xcpt:Receiver> …</xcpt:Receiver>
            <xcpt:RequestDetails>
                                <xcpt:RequestID class="0">20110228-11112222</xcpt:RequestID>
                                <xcpt:TimeStamp>2011-02-28T07:39:50</xcpt:TimeStamp>
                                   <xcpt:Application>…</xcpt:Application>
```

```
                    <xcpt:Procedure>http://www.extra-
standard.de/procedures/DEUEV</xcpt:Procedure>
                    <xcpt:DataType>http://www.extra-
standard.de/datatypes/Sofortmeldung</xcpt:DataType>
                    <xcpt:Scenario>scenario/request-with-acknowledgement</xcpt:Scenario>
        </xcpt:RequestDetails>
    </xreq:TransportHeader>

    <xreq:TransportPlugIns>
        <xplg:DataSource version="1.0">
                    <xplg:DataContainer type="http://www.extra-standard.de/container/FILE"
                    name="EDUA000003" created="2011-02-27T15:08:59" encoding="I8"/>
    …..</xplg:DataSource>
    </xreq:TransportPlugIns>

    <xreq:TransportBody> ….</xreq:TransportBody>
</xreq:XMLTransport>
```

Damit muss die Anforderung zur Wiederholung der Response mittels RepeatResponse folgendermaßen formuliert werden:

```
<xreq:Transport version=”1.3” …>
    <xreq:TransportHeader>
        <xcpt:TestIndicator> …</xcpt:TestIndicator>
        <xcpt:Sender> …</xcpt:Sender>
        <xcpt:Receiver> …</xcpt:Receiver>
        <xcpt:RequestDetails>
                    <xcpt:RequestID class="0">20110301-11117777</xcpt:RequestID>
                    <xcpt:TimeStamp>2011-03-01T09:49:51</xcpt:TimeStamp>
                    <xcpt:Application>…</xcpt:Application>

                    <xcpt:Procedure>DistributionServer</xcpt:Procedure>
                    <xcpt:DataType>RepeatResponse</xcpt:DataType>
                    <xcpt:Scenario>scenario/request-with-response</xcpt:Scenario>
        </xcpt:RequestDetails>
    </xreq:TransportHeader>

    <xreq:TransportPlugIns>
        <xplg:DataTransforms version="1.1">
            <!— falls das DFÜ-Protokoll http verwendet wird, sollte zur Sicherheit unbedingt die
                fachlichen Daten im TransportBody verschlüsselt werden
                => erfordert <Base64CharSequence>; ansonsten genügt <ElementSequence>   -->
    …..</xplg:DataTransforms>
    </xreq:TransportPlugIns>

    <xreq:TransportBody>
        <xcpt:Data>
            <xcpt:Base64CharSequence>
                <xreq:XMLTransport version=”1.1” ….>
                    <xreq:TransportHeader>   <!— urspruenglicher TransportHeader -->
                    <xcpt:TestIndicator> …</xcpt:TestIndicator>
                    <xcpt:Sender> …</xcpt:Sender>
                    <xcpt:Receiver> …</xcpt:Receiver>
                    <xcpt:RequestDetails>
                    <xcpt:RequestID class="0">20110228-11112222</xcpt:RequestID>
```

```
                                <xcpt:TimeStamp>2011-02-28T07:39:50</xcpt:TimeStamp>
                                <xcpt:Application>…</xcpt:Application>
                                <xcpt:Procedure>http://www.extra-
standard.de/procedures/DEUEV</xcpt:Procedure>
                                <xcpt:DataType>http://www.extra-
standard.de/datatypes/Sofortmeldung</xcpt:DataType>
                                <xcpt:Scenario>scenario/request-with-acknowledgement</xcpt:Scenario>
                                    </xcpt:RequestDetails>
                        </xreq:TransportHeader>

                        <xreq:TransportPlugIns>
                                    <xplg:DataSource version="1.0">
                                      <xplg:DataContainer type="…" name="EDUA000003" … />
                                    </xplg:DataSource>
                        </xreq:TransportPlugIns>
                        <xreq:TransportBody/>
                    </xreq:XMLTransport>
                </xcpt:Base64CharSequence>
            </xcpt:Data>
        </xreq:TransportBody>
</xreq:XMLTransport>
```

Die Response – die Antwort – auf die Standardnachricht RepeatResponse sieht z.B. folgendermaßen aus:

```
<xres:Transport version=”1.3” …>
        <xres:TransportHeader>
            <xcpt:TestIndicator> …</xcpt:TestIndicator>
            <xcpt:Sender> …</xcpt:Sender>
            <xcpt:Receiver> …</xcpt:Receiver>
            <xcpt:RequestDetails> … <!-- neuer Request: neue Angaben der RequestDetails -->
                                <xcpt:RequestID class="0">20110301-11117777</xcpt:RequestID>
                                <xcpt:TimeStamp>2011-03-01T09:49:51</xcpt:TimeStamp>
                                  <xcpt:Application>…</xcpt:Application>

                                  <xcpt:Procedure>DistributionServer</xcpt:Procedure>
                                  <xcpt:DataType>RepeatResponse</xcpt:DataType>
                                  <xcpt:Scenario>scenario/request-with-response</xcpt:Scenario>
            </xcpt:RequestDetails>
            <xcpt:ResponseDetails> … <!-- direkte Antwort auf den neuen Request -->
                                <xcpt:ResponseID>…</xcpt:ResponseID>
                                <xcpt:TimeStamp>2011-03-01-19T09:50:07Z</xcpt:TimeStamp>
                                <xcpt:Report highestWeight="http://www extra-standard de/weight/OK">
                                        <xcpt:Flag weight="http://www extra-standard de/weight/OK">
                                                <xcpt:Code>T000</xcpt:Code>
                                                <xcpt:Text>Alles OK</xcpt:Text>
                                        </xcpt:Flag>
                                </xcpt:Report>
                            </xcpt:ResponseDetails>
        </xres:TransportHeader>
        <xres:PackagePlugIns>
            <xplg:DataTransforms version="1.1"> ….</xplg:DataTransforms>
        </xres:PackagePlugIns>
        <xres:TransportBody>
            <xcpt:Data>
                <xcpt:Base64CharSequence>
                        <xres:XMLTransport version=”1.1” ….>
```

```
                    <xres:TransportHeader>
                    <!-- hier kommt der TransportHeader wie er beim Sender hätte ankommen sollen→
                            <xcpt:TestIndicator> …</xcpt:TestIndicator>
                            <xcpt:Sender> …</xcpt:Sender>
                            <xcpt:Receiver> …</xcpt:Receiver>
                            <xcpt:RequestDetails>
                                <xcpt:RequestID class="0">20110228-11112222</xcpt:RequestID>
                                <xcpt:TimeStamp>2011-02-28T07:39:50Z</xcpt:TimeStamp>
                                <xcpt:Application>…</xcpt:Application>
                                <xcpt:Procedure>http://www.extra-
standard.de/procedures/DEUEV</xcpt:Procedure>
                                <xcpt:DataType>http://www.extra-
standard.de/datatypes/Sofortmeldung</xcpt:DataType>
                                <xcpt:Scenario>scenario/request-with-
acknowledgement</xcpt:Scenario>
                            </xcpt:RequestDetails>
                            <xcpt:ResponseDetails>
                            …..<xcpt:ResponseID>…</xcpt:ResponseID>
                            …..<xcpt:TimeStamp>2011-02-28-T07:40:01Z</xcpt:TimeStamp>
                                <xcpt:Report highestWeight="http://www extra-standard de/weight/OK">
                                        <xcpt:Flag weight="http://www extra-standard de/weight/OK">
                                                <xcpt:Code>T000</xcpt:Code>
                                                <xcpt:Text>Alles OK</xcpt:Text>
                                        </xcpt:Flag>
                                </xcpt:Report>
                            </xcpt:ResponseDetails>
        ……          </xres:TransportHeader>
        ……          </xres:TransportBody>
                </xres:XMLTransport>
            </xcpt:Base64CharSequence>
        </xcpt:Data>
    <xres:TransportBody/>
</xres:XMLTransport>
```

Die Antwort zeigt in diesem Beispiel, dass die ursprüngliche Datensendung korrekt angenommen und weitergeleitet wurde, siehe ResponseDetails in der wiederholten Response.

2.3. Die neue eXTra-Standardnachricht StatusRequest

Die neue Standardnachricht **StatusRequest** kann für die Spezifikation, auf welche Sendung(en) sich die gewünschte Auskunft bezieht, auf die Sprachmittel der bereits bestehenden Standardnachricht DataRequest – genauer auf das Element **Query** – zurückgreifen. Da man mit **StatusRequest** auch eine Menge von Stati anfordern kann (z.B. mit <Argument property="RequestID"> und <GT>), ist es sinnvoll, die Menge an zurück zu liefernden Stati begrenzen zu können. Auch dafür kann man Anleihen bei der Standardnachricht DataRequest, nämlich beim Element **Control,** nehmen.

Damit ergibt sich folgende Grobstruktur:

<StatusRequest version="1.0">
    <Query> ……</Query>
    <Control> … </Control>
</StatusRequest>

2.3.1. Das Element Query

Das Element Query könnte analog zur Standardnachricht DataRequest eine Folge von n Kindelementen **Argument** enthalten. **Argument** selbst kennt drei Attribute (@event, @property und @type) und hat als alternative Kindelemente EQ, GE, LE, GT, LT. Bei den beiden Attributen @event und @property des Elementes **Argument** kann man auf die bestehenden Codelisten EventNames und DataRequestPropertyNames zurückgreifen.

Codeliste EventNames

	xmsg:EventNamesType
Inhalt	Identifikatoren von Ereignissen, die im Ablauf eines eXTra Kommunikationsszenarios auftreten können
Datentyp	`xsd:anyURI`
Individualisierbar	Nein
Beliebige Domain	Ja

Vordefinierte Werte:

`http://www.extra-standard.de/event/SendData`
`http://www.extra-standard.de/event/RequestData`

Codeliste DataRequestPropertyNames

	xmsg:DataRequestPropertyNamesType
Inhalt	Identifikatoren abfragbare Eigenschaften
Datentyp	xsd:anyURI
Individualisierbar	Nein
Beliebige Domain	Nein

Vordefinierte Werte:

```
http://www.extra-standard.de/property/SenderID
http://www.extra-standard.de/property/ReceiverID
http://www.extra-standard.de/property/Procedure
http://www.extra-standard.de/property/DataType
http://www.extra-standard.de/property/ResponseID
http://www.extra-standard.de/property/ResponseCreationTimeStamp
http://www.extra-standard.de/property/ResponseFileName
http://www.extra-standard.de/property/Layer
```

Diese Codeliste sollte man noch erweitern, um die Begriffe des Senders verwenden zu können und – besonders für die Verfahren der GKV und der Rentenversicherung – den Dateinamen im PlugIn DataSource als Argument verwenden zu können. Damit ergäbe sich die neue Codeliste StatusRequestPropertyNames.

Codeliste StatusRequestPropertyNames

	xmsg:StatusRequestPropertyNamesType
Inhalt	Identifikatoren abfragbare Eigenschaften
Datentyp	xsd:anyURI
Individualisierbar	Nein
Beliebige Domain	Nein

Vordefinierte Werte:

```
http://www.extra-standard.de/property/SenderID
http://www.extra-standard.de/property/ReceiverID
http://www.extra-standard.de/property/Procedure
http://www.extra-standard.de/property/DataType
http://www.extra-standard.de/property/RequestID
http://www.extra-standard.de/property/ResponseID
http://www.extra-standard.de/property/RequestCreationTimeStamp
http://www.extra-standard.de/property/ResponseCreationTimeStamp
http://www.extra-standard.de/property/RequestFileName
http://www.extra-standard.de/property/Layer
```

2.3.2. Das Element Control

Das Element **Control** kann man analog zur Standardnachricht DataRequest definieren und an den Kontext der neuen Standardnachricht **StatusRequest** anpassen, indem das Kindelement **MaximumResults** eingeführt wird. Alternativ zu MaximumResults kann auch MaximumSize verwendet werden.

2.3.3. Beispiel für die neue Standardnachricht StatusRequest

Es werden die Stati aller Lieferungen/Packages/Messages angefordert, die der angegebenen Selektion entsprechen.
Argument/@property gibt die abzufragende Eigenschaft an. Das Kindelement EQ, GT usw. spezifiziert Vergleichsoperator und Vergleichswert.

RequestID / ResponseID bzw. RequestFilename / ResponseFilename sind Alternativen, um den Range der ursprünglichen Lieferungen präzise einzugrenzen, von denen man den aktuellen Status anfordert

Voraussetzung für die Nutzung der RequestID / ResponseID / RequestFilename / ResponseFilename für die Spezifikation einer Menge ursprünglicher Sendungen ist, dass der bzw. die Begriffe streng aufsteigend eineindeutig vergeben werden.

```
<xmsg:StatusRequest>
    <xmsg:Query>
        <xmsg:Argument property="http://www.extra-standard.de/property/ReceiverID" type="xs:string">
                <xmsg:EQ>7259</xmsg:EQ>
        </xmsg:Argument>
        <xmsg:Argument property="http://www.extra-standard.de/property/Procedure" type="xs:string">
                <xmsg:EQ>DUA</xmsg:EQ>
        </xmsg:Argument>
        <xmsg:Argument property="http://www.extra-standard.de/property/Datatype" type="xs:string">
                <xmsg:EQ>Meldung</xmsg:EQ>
        </xmsg:Argument>
        <xmsg:Argument property="http://www.extra-standard.de/property/RequestID" type="xs:string">
                <xmsg:GT>009999</xmsg:GT>
        </xmsg:Argument>
    </xmsg:Query>
```

```
<!--
        Alternativen zu RequestID sind: ResponseID, RequestFilename oder ResponseFilename.
```

Ein Begriff aus der Sphäre des Empfängers, z.B. die ResponseID, ist nur dann eine Alternative, wenn die eXTra-Response auf den ursprünglichen eXTra-Request beim Sender tatsächlich auch erfolgreich empfangen wurde.

```
        <xmsg:Argument property="http://www.extra-standard.de/property/ResponseID" type="xs:string">
                <xmsg:GT>7259</xmsg:GT>
        </xmsg:Argument>

        <xmsg:Argument property="http://www.extra-standard.de/property/DataSourceFilename"
type="xs:string">
                <xmsg:GT>EDUA0000003</xmsg:GT>
        </xmsg:Argument>
-->

        <xmsg:Control>
                <xmsg:MaximumResults>100</xmsg:MaximumResults>
                <!--    Begrenzung der Anzahl rueck zu meldender Stati auf 100 … →
        </xmsg:Control>

</xmsg:StatusRequest>
```

2.4. Die neue eXTra-Standardnachricht StatusResponse

Die neue Standardnachricht für Rückmeldungen – Vorschlag für den Namen ist **StatusResponse** – besteht aus zwei Elementen: Dem Element **Property**, mit dem die ursprüngliche Lieferung identifiziert und deren Status mit dem zweiten Element **Trace** exakt mitgeteilt wird.

Damit ergibt sich folgende Grobstruktur für die neue Standardnachricht **StatusResponse:**

```
<StatusResponse version="1.0">
        <Property> ……</Property>
        <Trace>……</Trace>
</StatusResponse>
```

2.4.1. Das Element Property

Das Element **Property** ist von der Standardnachricht ConfirmationOfReceipt her bereits bekannt und hat den gleichen Aufbau; es kann mehrfach vorkommen, um eine leichte Lesbarkeit und Eindeutigkeit der Sendung sicherzustellen.

Property hat demgemäß drei Attribute (@name, @type und @event) und kennt als Kindelement nur das Element **Value**.

Damit ergibt sich folgende bekannte Grobstruktur für das Element **Property:**

```
<Property name="…" type="… event=…">
        <Value>…..</Value>
</Property>
…..
```

Bei den Attributen @name und @event des Elements **Property** kann man auf die Codelisten StatusRequestPropertyNames und EventNames zurückgreifen (siehe oben unter 2.3.1).

2.4.2. Das Element Trace

Das Element **Trace** selbst hat nur ein Kindelement, nämlich **Checkpoint. Checkpoint** könnte mehrfach auftreten, wenn die durchlaufene Prozesskette auf Empfängerseite aufgezeigt werden soll.

Damit ergibt sich für den allgemeinen Fall folgende Grobstruktur für das Element **Trace:**

```
<Trace>
        <Checkpoint>
                <Layer>…</Layer>
                <Timestamp>…</Timestamp>
                <Status>…</Status>
                <Report highestWeight=”….”t>
                        <Flag weight=”….”>
                                <Code>…</Code>
                                <Text>...</Text>
                        </Flag>
                </Report>
        </Checkpoint>
</Trace>
```

Das Element **Checkpoint** hat insgesamt vier Kindelemente, nämlich die drei Pflichtelemente **Layer**, **Timestamp** und **State** sowie das optionale Element **Report.**

Das Element Checkpoint

Die zugelassenen Werte des Elementes **Layer** sind wie folgt:

Codeliste LayerNames

	xmsg:LayerNamesType
Inhalt	Identifikatoren abfragbare Eigenschaften
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Nein

Vordefinierte Werte:

```
http://www.extra-standard.de/layer/Transport
http://www.extra-standard.de/layer/Package
http://www.extra-standard.de/layer/Message
http://www.extra-standard.de/layer/Application
http://www.extra-standard.de/layer/Delivery
```

Die vordefinierten Werte für das Element **Layer** sind bereits für einen möglichen Vollausbau ausgelegt, bei dem auch Stati in Erfahrung gebracht werden können, die das Gesamtverfahren

umfasst, bestehend aus Sendeprozess, Fachverfahren, Abhol- und Bestätigungsprozess, bzw. die das betroffene Fachverfahren definiert (über eine Individualisierung des vordefinierten Wertes für layer/Application).

Eine erste einfache Ausbaustufe könnte z.B. lediglich Stati zurückmelden, die nur den Sendeprozess umfassen (z.B. weil das zugehörige Fachverfahren noch keine Stati zum Ver-/Bearbeitungsstand einer gesendeten Lieferung melden kann). Im Rahmen des Profilierungsvorgangs könnte die Standardnachricht StatusResponse entsprechend profiliert werden, indem die Codeliste für LayerNames nur die Werte für Transport (und ggfls für Package und Message) enthält.

Das Element **Timestamp** ist bezüglich der zugelassenen Werte so definiert wie das Element **Timestamp** des eXTra Basisschemas, nämlich als Datentyp DateTime.

Für das Element **Status** kann folgende Codeliste definiert werden:

Codeliste StatusNames

	`xcode:StatusNamesType`
Inhalt	Identifikatoren für einen Zustand auf Empfängerseite
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Nein

Vordefinierte Werte:

```
http://www.extra-standard.de/status/ACCEPTED
http://www.extra-standard.de/status/PROCESSING
http://www.extra-standard.de/status/COMPLETED
http://www.extra-standard.de/status/CONFIRMED
http://www.extra-standard.de/status/FAILED
```

Die vordefinierten Werte für das Element **Status** sind bereits für einen möglichen Vollausbau ausgelegt, die das Gesamtverfahren umfasst (Sendeprozess, Fachverfahren, Abhol- und Bestätigungsprozess) und bei dem auch differenzierte Bearbeitungsstufen eines Prozessschrittes (einer eXTra-Ebene bzw. des Fachverfahrens) in Erfahrung gebracht werden können.

Eine erste Ausbaustufe könnte z.B. lediglich Bearbeitungsstufen innerhalb von eXTra-Ebenen zurückmelden, die nur beim Sendeprozess auf Empfängerseite durchlaufen werden (z.B. weil das zugehörige Fachverfahren keine Verbindung herstellen kann zur ursprünglicher Lieferung mit den entsprechenden eXTra-Begriffen RequestID/ResponseID oder RequestFileName). Im Rahmen des Profilierungsvorgangs könnte die Standardnachricht StatusResponse entsprechend profiliert werden, indem die Codeliste für StatusNames nur die Werte für ACCEPTED, PROCESSING, COMPLETED und FAILED enthält.

Die Bedeutung der verschiedenen Ausprägungen ist wie folgt:

- „accepted“: Die übermittelten Daten sind auf der entsprechenden eXTra-Ebene bzw. beim Fachverfahren (siehe **Layer**) angekommen und zumindest in die lokale Datenhaltung übernommen worden. Weitere Arbeitsschritte, wie z.B. Validierung, Komprimieren/Dekomprimieren, Verschlüsseln/Entschlüsseln, Signieren/Signatur prüfen oder verarbeiten sind noch nicht angelaufen. Die Daten sind jedoch definitiv noch nicht an die nächste eXTra-Ebene oder an das Fachverfahren (siehe **Layer**) weitergereicht worden.
- „processing“: Die übermittelten Daten sind auf der entsprechenden eXTra-Ebene bzw. beim Fachverfahren (siehe **Layer**) angekommen und werden dort gerade be- oder verarbeitet. Die notwendigen Arbeitsschritte auf einer eXTra-Ebene, wie z.B. Validierung, Komprimieren/Dekomprimieren, Verschlüsseln/Entschlüsseln, Signieren/Signatur prüfen usw. bzw. das Bearbeiten durch das Fachverfahren werden gerade durchlaufen, Fehler sind bisher nicht aufgetreten. Die Daten wurden noch nicht an die nächste eXTra-Ebene bzw. an das Fachverfahren (siehe **Layer**) weitergereicht.
- „completed“: Die jeweilige eXTra-Ebene oder das Fachverfahren hat die Be- oder Verarbeitung erfolgreich abgeschlossen und die Daten erfolgreich an die nächste eXTra-Ebene oder an das Fachverfahren weitergereicht, bzw. das Fachverfahren hat die Daten erfolgreich verarbeitet und die dazugehörige Ergebnismeldung an den eXTra Delivery Server weitergereicht. Es traten keine Fehler auf.
- „confirmed“: Der Sender hat die dazugehörige Ergebnismeldung mittels DataRequest vom eXTra Delivery Server angefordert und den Erhalt mittels ConfirmationOfReceipt bestätigt. Es traten keine Fehler auf.
- „failed“: Auf der jeweiligen eXTra-Ebene oder im Fachverfahren ist ein Fehler aufgetreten.

Das Element Report

Das Element **Report** und dessen Kindelement **Flag** ist jeweils genauso wie im eXTra Basisstandard definiert. **Report** hat demgemäß ein Attribut **@highestWeight, Flag** das Attribut **@weight** und die beiden Kindelemente **Code** und **Text.**

Für das Attribut **@highestWeight** und **@weight** ist somit folgende Codeliste definiert:

Codeliste WeightCode

	xcode:WeightCodeType
Inhalt	Identifikatoren für Fehlergewichte
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Nein

Vordefinierte Werte:

```
http://www.extra-standard.de/weight/OK
http://www.extra-standard.de/weight/INFO
http://www.extra-standard.de/weight/WARNING
http://www.extra-standard.de/weight/ERROR
```

2.5. Die neue eXTra-Standardnachricht ListOfStatusResponse

Diese neue Standardnachricht ist dann vorgesehen, wenn die Standardnachricht RequestStatus den Status mehrerer Sendungen angefordert hat. Wie üblich für die ListOf-Nachrichten ist ListOfStatusResponse lediglich eine Klammer für eine Folge von StatusResponse Nachrichten.

Damit ergibt sich folgende Grobstruktur für die neue Standardnachricht **ListOfStatusResponse**

```
<ListOfStatusResponse version="1.0">
    <StatusResponse> ……</StatusResponse>
    …..
    <StatusResponse> ……</StatusResponse>
</ListOfStatusResponse>
```

2.6. Beispiele für die neue Standardnachricht StatusResponse und ListOfStatusResponse

Beispiel 1 für das Szenario 2

Der Sender hat die eXTra Standardnachricht StatusRequest abgegeben mit der Aufforderung, den Status der abgegebenen Lieferungen mit der RequestID gleich 009999 zurückzugeben. Die Lieferung besteht nur aus der eXTra Transport-Ebene und die Lieferung wurde vom eXTra Destribution-Server erfolgreich an das Fachverfahren weitergegeben, aber dort noch nicht weiter verarbeitet bzw. das Fachverfahren wurde in das Gesamtsystem noch nicht integriert (Minimalausbau).

Der TransportBody der Rückmeldung hat hierfür folgende Grobstruktur:

```
<xres:TransportBody>
    <xcpt:Data>
        <xcpt:ElementSeqence>
                <xmsg:StatusResponse> …. </xmsg:StatusResponse>
        </xcpt:ElementSeqence>
    </xcpt:Data>
</xres:TransportBody>
```

Die StatusResponse Nachricht sieht in diesem Fall z.B. folgendermaßen aus:

```
<xmsg:StatusResponse>

        <xmsg:Property name="http://www.extra-standard.de/property/ReceiverID" type="xs:string">
                <xmsg:Value>7259</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/Procedure" type="xs:string">
                <xmsg:Value>DEUEV</xmsg:Value>
        </xmsg:Propertyt>

        <xmsg:Property name="http://www.extra-standard.de/property/Datatype" type="xs:string">
                <xmsg:Value>Sofortmeldung</xmsg:Value>
        </xmsg:Argument>

        <xmsg:Property name="http://www.extra-standard.de/property/RequestID" type="xs:string">
                <xmsg:Value>009999</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/DatasourceFilename"
type="xs:string">
                <xmsg:Value>EDUA0000003</xmsg:Value>
        </xmsg:Property>
```

```
        <xmsg:Trace
                <msg:Checkpoint>
                        <msg:Layer>http://www.extra-standard.de/level/Transport</msg:Layer>
                        <msg:TimeStamp>2010-10-08T20:10:44</msg:TimeStamp>
                        <msg:Status>http://www.extra-standard.de/state/COMPLETED</msg:Status>
                        <msg:Report highestWeight=http://www.extra-standard.de/weight/INFO>
                                <msg:Flag weight=http://www.extra-standard.de/weight/INFO>
                                        <msg:Code>T1000</msg:Code>
                                        <msg:Text>alles ok</msg:Text>
                                </msg:Flag>
                        </msg:Report>
                </msg:Checkpoint>

        </xmsg:Trace>
</xmsg:StatusResponse>
```

Beispiel 2 für das Szenario 2

Der Sender hat die eXTra Standardnachricht StatusRequest abgegeben mit der Aufforderung, den Status aller abgegebenen Lieferungen mit der RequestID größer 009999 zurückzugeben.
Annahmen:

- Die Lieferungen bestehen immer nur aus der eXTra Transport-Ebene,
- die Statusanforderung trifft für drei Lieferungen zu,
- das Gesamtsystem auf Empfängerseite ist vollständig ausgebaut, d.h. eine vom Sender abgegebene Lieferung kann bis zuletzt (bis zur Bestätigung durch den Sender) mittels ConfirmationOfReceipt nachvollzogen werden,
- die älteste Lieferung (RequestID=009999) wurde vom Fachverfahren verarbeitet und die Verarbeitungsquittung an den eXTra Delivery-Server weitergeben, aber noch nicht mittels DataRequest abgeholt,
- die nächste Lieferung (RequestID=010000) wurde vom Fachverfahren verarbeitet, aber die Verarbeitungsquittung noch nicht an den eXTra Delivery-Server weitergegeben,
- die jüngste Lieferung (RequestID=010001) lief noch vor der Weiterleitung an das Fachverfahren auf einen Fehler auf der Transportebene.

Die Rückmeldung besteht aus der ListOfStatusResponse Nachricht, die wiederum aus einer Folge von StatusResponse Nachrichten besteht, z.B. aus drei Nachrichten. Der TransportBody hat hierfür folgende Grobstruktur:

```
<xres:TransportBody>
    <xcpt:Data>
        <xcpt:ElementSeqence>
            <xmsg:ListOfStatusResponse version=”1.0”>
                <xmsg:StatusResponse> …. </xmsg:StatusResponse>  <!-- Status zu Lieferung 1 -->

                <xmsg:StatusResponse> …. </xmsg:StatusResponse>  <!-- Status zu Lieferung 2 -->

                <xmsg:StatusResponse> …. </xmsg:StatusResponse>  <!-- Status zu Lieferung 3 -->
            </xmsg:ListOfStatusResponse>
        </xcpt:ElementSeqence>
    </xcpt:Data>
</xres:TransportBody>
```

Im Detail haben die drei Statusnachrichten z.B. folgende Ausprägung:

Status zu Lieferung 1 mit der RequestID 009999, fehlerfrei

```
<xmsg:StatusResponse>
        <xmsg:Property name="http://www.extra-standard.de/property/ReceiverID" type="xs:string">
                <xmsg:Value>7259</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/Procedure" type="xs:string">
                <xmsg:Value>DEUEV</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/Datatype" type="xs:string">
                <xmsg:Value>Sofortmeldung</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/RequestID" type="xs:string">
                <xmsg:Value>009999</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/DatasourceFilename"
type="xs:string">
                <xmsg:Value>EDUA0000018</xmsg:Value>
        </xmsg:Property>


        <xmsg:Trace
            <msg:Checkpoint>
                <msg:Layer>http://www.extra-standard.de/level/Delivery</msg:Layer>
                <msg:TimeStamp>2010-10-08T20:10:44</msg:TimeStamp>
                <msg:Status>http://www.extra-standard.de/state/ACCEPTED</msg:Status>
                <msg:Report highestWeight=http://www.extra-standard.de/weight/INFO>
                        <msg:Flag weight=http://www.extra-standard.de/weight/INFO>
                                <msg:Code>D1000</msg:Code>
                                <msg:Text>alles ok</msg:Text>
                        </msg:Flag>
                </msg:Report>
            </msg:Checkpoint>
        </xmsg:Trace>

</xmsg:StatusResponse>
```

Status zu Lieferung 2 mit der RequestID 010000, fehlerfrei

```
<xmsg:StatusResponse>

        <xmsg:Property name="http://www.extra-standard.de/property/ReceiverID" type="xs:string">
                <xmsg:Value>7259</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/Procedure" type="xs:string">
                <xmsg:Value>DEUEV</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/Datatype" type="xs:string">
                <xmsg:Value>Sofortmeldung</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/RequestID" type="xs:string">
                <xmsg:Value>010000</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/DatasourceFilename"
type="xs:string">
                <xmsg:Value>EDUA0000019</xmsg:Value>
        </xmsg:Property>


        <xmsg:Trace
            <msg:Checkpoint>
                <msg:Layer>http://www.extra-standard.de/level/Application</msg:Layer>
                <msg:TimeStamp>2010-10-09T20:15:14</msg:TimeStamp>
                <msg:Status>http://www.extra-standard.de/state/PROCESSING</msg:Status>
                <msg:Report highestWeight=http://www.extra-standard.de/weight/INFO>
                        <msg:Flag weight=http://www.extra-standard.de/weight/INFO>
                                <msg:Code>A0000</msg:Code>
                                <msg:Text>alles ok</msg:Text>
                        </msg:Flag>
                </msg:Report>
            </msg:Checkpoint>
        </xmsg:Trace>



</xmsg:StatusResponse>
```

Status zu Lieferung 3 mit der RequestID 010001; Fehler auf der Transportebene

```
<xmsg:StatusResponse>

        <xmsg:Property name="http://www.extra-standard.de/property/ReceiverID" type="xs:string">
                <xmsg:Value>7259</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/Procedure" type="xs:string">
                <xmsg:Value>DEUEV</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/Datatype" type="xs:string">
                <xmsg:Value>Sofortmeldung</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/RequestID" type="xs:string">
                <xmsg:Value>010001</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property name="http://www.extra-standard.de/property/DatasourceFilename"
type="xs:string">
                <xmsg:Value>EDUA0000020</xmsg:Value>
        </xmsg:Property>


        <xmsg:Trace
            <msg:Checkpoint>
                <msg:Layer>http://www.extra-standard.de/level/Transport</msg:Layer>
                <msg:TimeStamp>2010-10-09T21:05:04</msg:TimeStamp>
                <msg:Status>http://www.extra-standard.de/state/FAILED</msg:Status>

                <msg:Report highestWeight=http://www.extra-standard.de/weight/ERROR>
                        <msg:Flag weight=http://www.extra-standard.de/weight/ERROR>
                                <msg:Code>TF100</msg:Code>
                                <msg:Text>TransportEbene Fehler: Die Nachricht konnte nicht
                                        entschluesselt werden</msg:Text>
                        </msg:Flag>
                </msg:Report>
            </msg:Checkpoint>
        </xmsg:Trace>


</xmsg:StatusResponse>
```