einheitliches XML-basiertes Transportverfahren

Standardnachrichten 1.5
Schnittstellenbeschreibung

Ausgabestand 1.5.0

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.

Änderungsprotokoll für Version 1.5.0. Ausgabestand 1.5.0

Autor[en]	Datum	Beschreibung
[gelöscht]	05.06.2018	Aufnahme von `RepeatResponseRequest`

Änderungsprotokoll für Version 1.4. Ausgabestand 1.4.0

Autor[en]	Datum	Beschreibung
[gelöscht]	25.06.2014	Erstellung 1.4 auf Basis Final 1.3 Codelisten `DataRequestPropertyNamesType` & `Property-NamesType` um die Properties `Request*` und `State` erweitert. Codeliste `ListRequestPropertyNamesType` `DataRequestArgumentType` und `StatusRequestArgu-mentType` um das Attribut `order` erweitert Nachricht `ListRequest` Nachricht `ListResponse` Erweiterte Nachricht `DataRequestV1.3` und `StatusRequest V1.1`

Änderungsprotokoll für Version 1.3. Ausgabestand 1.3.0

Autor[en]	Datum	Beschreibung
[gelöscht]	17.11.2011	Erstellung 1.3 auf Basis Final 1.2 Ausgabestand 1.2.1 Codelisten `DataRequestPropertyNamesType` & `Property-NamesType` um die Properties `Request*` und `Layer` erweitert sowie Properties aus beliebigen Domains zugelassen. Codeliste `StatusRequestPropertyNamesType` Nachricht `StatusRequest` Nachricht `StatusResponse` Nachricht `ListOfStatusResponse` Nachricht `RepeatResponse`

Änderungsprotokoll für Version 1.2. Ausgabestand 1.2.1

Autor[en]	Datum	Beschreibung
[gelöscht]	29.07.2010	Erstellung 1.2 auf Basis Final 1.1 `PropertySet`: Attribut `event` Nachricht `ListOfConfirmationOfReceipt` Nachricht `ListOfDataRequest`

Änderungsprotokoll für Version 1.1. bis 1.1.1

Autor[en]	Datum	Beschreibung
[gelöscht]	01.11.2009	Erstellung 1.1
	08.03.2010	Freigabe

Inhaltsverzeichnis

1 Einleitung 6
1.1 Ausgabestand 6
1.2 Standardnachrichten 6
1.3 Begriffe 8
2 Layout und Notationen 9
2.1 Referenzen 9
2.2 Formatierungen 9
2.3 Beschreibung der Regeln 9
2.4 Beschreibung der Elemente und Attribute 10
2.5 Graphische Symbole 11
3 Aufbau von Standardnachrichten [normativ] 14
3.1 Namensräume und Präfixe 14
4 eXTra Codelisten [normativ] 15
4.1 DataRequestPropertyNames 16
4.2 EventNames 16
4.3 ExtraLayerNames 17
4.4 ListRequestPropertyNames 17
4.5 ListResponsePropertyNames 18
4.6 PropertyNames 19
4.7 RequestState 19
4.8 SortingOrder 20
4.9 StatusNames 20
4.10 StatusRequestPropertyNames 21
4.11 WeightCodes 21
4.12 XSDTypeCodes 22
5 Nachricht ConfirmationOfReceipt [normativ] 24
5.1 Beispieldokument 25
5.2 Element ConfirmationOfReceipt 26
5.3 Element Property 28
5.4 Element PropertySet 31
5.5 Element Value 32
6 Nachricht DataRequest [normativ] 33
6.1 Beispieldokument 34
6.2 Element DataRequest 36
6.3 Element Argument 38
6.4 Element Control 41
6.5 Element EQ 42
6.6 Element GE 43
6.7 Element GT 44
6.8 Element IN 45
6.9 Element LE 46
6.10 Element LT 47
6.11 Element MaximumMessages 48
6.12 Element MaximumPackages 49
6.13 Element MaximumResults 50
6.14 Element MaximumSize 51
6.15 Element Query 52
7 Nachricht ListRequest [normativ] 53
7.1 Beispieldokument 54
7.2 Element ListRequest 55
7.3 Element Argument 56

7.4 Element Query....59
8 Nachricht ListResponse [normativ]....60
8.1 Beispieldokument....61
8.2 Element ListResponse....63
8.3 Element Entry....63
9 Nachricht ListOfConfirmationOfReceipt [normativ]....65
9.1 Beispieldokument....66
9.2 Element ListOfConfirmationOfReceipt....69
10 Nachricht ListOfDataRequest [normativ]....70
10.1 Beispieldokument....71
10.2 Element ListOfDataRequest....73
11 Nachricht ListOfStatusResponse [normativ]....74
11.1 Beispieldokument....75
11.2 Element ListOfStatusResponse....76
12 Nachricht RepeatResponse [normativ]....77
13 Nachricht RepeatResponseRequest....78
13.1 Beispieldokument....78
13.2 Element RepeatResponseRequest....79
13.3 Element Argument....80
13.4 Element Query....82
14 Nachricht StatusRequest [normativ]....83
14.1 Beispieldokument....84
14.2 Element StatusRequest....85
14.3 Element Argument....86
14.4 Element Query....89
15 Nachricht StatusResponse [normativ]....90
15.1 Beispieldokument....91
15.2 Element StatusResponse....92
15.3 Element Checkpoint....93
15.4 Element Code....94
15.5 Element Flag....95
15.6 Element Layer....96
15.7 Element Report....97
15.8 Element Status....98
15.9 Element Text....99
15.10 Element Timestamp....100
15.11 Element Trace....101
16 Ressourcen....102
16.1 XSD Schema Ressourcen....102
17 Referenzen....103

1 Einleitung

1.1 Ausgabestand

Gültigkeit und Versionierung: Dieses Dokument enthält die normative technische Spezifikation der Schnittstelle der eXTra-Standardnachrichten in der Version 1.5.0 des eXTra-Basis-Standards.

In der Versionsangabe dieses Dokuments V *n.m.r* ist in den ersten beiden Ziffern *n.m* die jeweilige Version (hier Version 1.5) der zugrundeliegenden Schnittstelle der eXTra-Standardnachrichten angeführt, während die letzte Ziffer *r* den jeweiligen Ausgabestand (hier Ausgabestand 0) wiedergibt.

Hinweis zum Gebrauch: Bibliographische Referenzen stehen in eckigen Klammern ([*bibref*]) und sind am Ende dieser Spezifikation beschrieben.

1.2 Standardnachrichten

eXTra, das e*inheitliche XML-basierte Transportverfahren*, ist ein gemeinschaftlich von Unternehmen und Behörden entwickelter, offener Standard für die Datenübermittlung. Eine kurze Einführung sowie eine detaillierte Beschreibung finden sich in [EINF] bzw. [KOMP]. Diese Dokumente und sämtliche öffentlichen Informationen über eXTra sind im Internet unter der Adresse http://www.extra-standard.de abrufbar.

Dieses Dokument enthält die normative technische Spezifikation der eXTra-*Standardnachrichten*. Standardnachrichten sind kein Teil der eXTra-Kommunikationsschicht, sondern als Nutzdaten zu betrachten und wie diese zu verarbeiten. Aus diesem Grund befinden sie sich in einem separaten Namensraum und haben keinerlei Abhängigkeit zu den Schemata der eXTra-Kommunikationsschicht. Sie standardisieren einige in vielen Anwendungsszenarien vorkommende Nachrichten wie z.B. Empfangsbestätigungen, und stellen damit eine weitere Vereinfachung bei der Einführung und Verwendung des eXTra-Standards dar, siehe auch „Überblick über die eXTra-Standardnachrichten“ [ÜMSG].

Diese Spezifikation beschreibt die Strukturen und Felder der eXTra-Standardnachrichten. Regeln und Empfehlungen für den Entwurf sowie die Implementierung eines Datenübermittlungssystems auf Basis von eXTra sind im Designleitfaden [DSIG] und in den Best Practices [BEST] beschrieben. Diese Dokumente bilden die Grundlage für den Entwurf und die Imple-

mentierung der eXTra-Kommunikationsschnittstelle und richten sich vor allem an Anwendungsarchitekten und Entwickler. Die Profilierung der eXTra-Kommunikationsschnittstelle ist in [PROF] erläutert.

eXTra-Standardnachrichten basieren auf der *Extensible Markup Language – XML* – in der Version 1.0, wie beschrieben durch [XML] und sind daher stets XML-Instanzen. Für die Beschreibung und Prüfung von XML-Instanzen durch sogenannte *Schemata* stehen verschiedene Sprachmittel zur Verfügung, von denen am häufigsten *XML Schema Definiton – XSD* – [XSD] zum Einsatz kommt. XSD wird von praktisch allen gängigen kommerziellen und nichtkommerziellen XML-Entwicklungswerkzeugen unterstützt. Aus diesem Grund wurden auch die eXTra-Standardnachrichten mit XSD-Schema definiert.

eXTra vermeidet die Verwendung von XSD-Sprachmitteln, die XML-Instanzen an eine Validierung mittels bestimmter XSD-Instanzen binden, wie zum Beispiel die Typsubstitution. Dies eröffnet Anwendern die Möglichkeit, eigene Schemata zu verwenden, solange diese die Schnittstellenspezifikation unter Beachtung der Profilierung korrekt implementieren. Eine Beschränkung auf XSD als Sprachmittel besteht dabei nicht. Das Schema ist in diesem Sinne als ein Bestandteil Teil der Schnittstellenimplementierung zu betrachten, welche vollständig nach außen gekapselt ist. Jedes Schema, das die Schnittstellenspezifikation korrekt implementiert, muss jede Nachrichteninstanz, die alle Regeln der Schnittstellenspezifikation erfüllt, als valide erkennen.

Diese Spezifikation beschreibt Elemente, Attribute und Daten weitgehend neutral in Bezug auf die möglichen Sprachmittel für die Erstellung von Schemadefinitionen. Lediglich für die Beschreibung primitiver Datentypen wie Zeichenketten, numerischen Werten und Datumsangaben werden XSD-Sprachmittel verwendet, da diese gut verständlich sind und auch in anderen Schemasprachen wie z.B. Relax NG [RELAX] Anwendung finden. In den graphischen Darstellungen der Elemente können schemaspezifische oder XSD-spezifische Konstrukte wie etwa Namen von Typdefinitionen erscheinen. Dies ist eine Folge der Generierung der Graphiken aus den eXTra-XSD-Instanzen, hat aber keine normative Wirkung.

Der normative Spezifikationsteil beginnt mit den einleitenden Kapiteln, Definitionen und Aufbau von eXTra-Standardnachrichten. Anschließend folgt die Spezifikation den Namensräumen [XNAM] der Elemente. Jeder Namensraum ist in einem separaten Kapitel beschrieben. Soweit in einem Namensraum ein Wurzelelement existiert, ist dieses als erstes beschrieben. Ansonsten erscheinen die Elemente in alphabetischer Reihenfolge.

Die Kapitel des normativen Spezifikationsteiles sind mit **[normativ]** gekennzeichnet.

1.3 Begriffe

In Anlehnung an [RFC2119] verwendet diese Spezifikation die unten aufgeführten Modalverben und Adjektive wie dargestellt in durchgängiger Großschreibung und mit der beschriebenen Bedeutung, wenn sie an der Verwendungsstelle eine normative Wirkung entfalten. Sie können innerhalb der mit **[normativ]** gekennzeichnetem Kapitel in allen konjugierten bzw. deklinierten Formen erscheinen.

MUSS / ERFORDERLICH: Die Anforderung ist zwingend und ohne Ausnahme zu erfüllen
DARF NICHT / VERBOTEN: Das Verbot ist zwingend und ohne Ausnahme einzuhalten
SOLL / EMPFOHLEN: Die Anforderung ist zu erfüllen, wenn nicht besondere technische oder fachliche Umstände eine Ausnahme begründen
SOLL NICHT / NICHT EMPFOHLEN: Das Verbot ist einzuhalten, wenn nicht besondere technische oder fachliche Umstände eine Ausnahme begründen
KANN / OPTIONAL: Die Anforderung ist vollständig optional und ihre Erfüllung daher in das Belieben des Implementierenden gestellt; eine Anwendung, die mit einer anderen Implementierung kommuniziert, MUSS sich darauf einstellen, dass diese nicht die betroffene Anforderung erfüllt.

Werden diese Begriffe in Kleinschreibung verwendet, sind die mit ihnen beschriebenen Eigenschaften oder Vorgänge entweder nicht Teil einer Regel, oder sie wiederholen an anderer Stelle des eXTra-Regelwerkes definierte Normen. Sie haben im zweiten Fall trotzdem keine normative Wirkung, da jede Regel innerhalb des eXTra-Regelwerkes nur an einer Stelle als Norm definiert ist.

2 Layout und Notationen

2.1 Referenzen

Bibliographische Referenzen stehen in eckigen Klammern ([*bibref*]) und sind am Ende dieser Spezifikation beschrieben.

2.2 Formatierungen

Es werden folgende Formatierungen verwendet:

Arial 11pt für den laufenden Text einschließlich bibliographischer Referenzen.

Arial 11pt kursiv für die erstmalige Erwähnung wichtiger Begriffe, sowie für fremdsprachliche Ausdrücke, soweit diese hervorgehoben werden sollen.

`Courier New 11pt` für XML-Komponenten wie Elemente und Attribute sowie für konstante bzw. vordefinierte Werte.

`Courier New 11pt` für kleinere Beispielfragmente aus XML-Instanzen.

`Courier New 8pt` für größere Beispielfragmente aus XML-Instanzen.

In Überschriften kann von diesen Formatierungen abgewichen werden.

2.3 Beschreibung der Regeln

Jede Regel des eXTra-Regelwerkes trägt eine eindeutige Regelbezeichnung und ist einem Regelbereich zugeordnet, der durch einen Buchstaben identifiziert wird:

Buchstabenkürzel	Regelbereich
I	*Implementierung*
P	*Profilierung*
X	*XML Schnittstelle*

In einem engen Zusammenhang stehende Regeln können zu einer Regelgruppe zusammengefasst sein. Regeln und Regelgruppen sind innerhalb eines Regelbereiches mit 1 beginnend durchnummeriert. Innerhalb einer Regelgruppe sind wiederum die Regeln durchnummeriert. Hieraus ergibt sich folgender Aufbau der Regelbezeichnung:

[*rb rn*] bzw.
[*rb rg.rn*]

wobei rb de Regelbereich, rg die Regelgruppe und rn die Regelnummer angibt. Beispiele:

[I 12], [P 8.3].

2.4 Beschreibung der Elemente und Attribute

Die Beschreibung eines Elementes und seiner Attribute setzt sich zusammen aus:

Einer Tabelle „Element“, i.d.R. mit grafischer Darstellung,
je einer Tabelle „Attribut“ pro Attribut,
einer Tabelle „Profilierung“, falls vorhanden,
einer textuellen Beschreibung.

Tabelle „Element“

	präfigierter_elementname
Inhalt	*kurzbeschreibung_element*
Datentyp	*inhaltstyp_element*
Benutzt von	*präfigierte_namen_elternelemente*

element_graphik

Im Einzelnen:

Inhalt: Eine kurze Beschreibung des Elementinhaltes.

Datentyp: Beschreibt den Inhalt des Elementes in typisierender Weise oder durch Angabe eines XSD-Datentyps (Attribute werden in einer separaten Tabelle beschrieben);

`##elements`: Das Inhaltsmodell des Elementes enthält mindestens ein weiteres Element und ggf. ein oder mehrere Attribute

`##empty`: Das Inhaltsmodell des Elementes sieht keinen Inhalt, aber ggf. ein oder mehrere Attribute vor

Wenn das Inhaltsmodell nur Daten als Inhalt vorsieht, wird der entsprechende *built-in* XSD-Datentyp angegeben.

Benutzt von: Die präfigierten Namen der Elemente, in denen das beschriebene Element verwendet wird, z.B. `xreq:MessageHeader`, `xres:MessageHeader`.

Tabelle „Attribut“

	attributname	*optionalität*
Inhalt	*kurzbeschreibung_attribut*	
Datentyp	*xsd-datentyp*	

Im Einzelnen:

attributname : Der Name des Attributes in XPath-Notation [XPATH], d.h. mit vorausgehendem @-Symbol

optionalität: Eines der Adjektive „erforderlich“ oder „optional“

Inhalt: Eine kurze textuelle Beschreibung des Attributes.

Datentyp: Ein einfacher XSD-Datentyp.

Tabelle „Profilierung“

Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
xcpt:Signatures	+	+	+		+	
xcpt:Data	*IMPLICIT*					

Diese Tabelle beschreibt für jede profilierbare Komponente des Elementes die Profilierungsoptionen, d.h. wie die Profilierung die Kardinaliät und/oder den Inhalt einer Komponente beeinflussen kann. Die Profilierungsoptionen sind als Folge von Spalten dargestellt; ein + in der durch eine Kombination von Komponente und Option bestimmten Zelle kennzeichnet die Option als zulässig. Bestimmte Komponenten werden implizit profiliert, d.h. ihre Profilierung leitet sich aus der Profilierung einer anderen Komponente, einer Funktion oder einer Struktur ab. In diesem Fall sind die Profilierungsspalten zusammengefasst und enthalten das Wort *IMPLICIT*.

Die Bezeichner bedeuten:

REQ: Profilierung kann die Komponente als erforderlich (required) deklarieren

OPT: Profilierung kann die Komponente als optional (optional) deklarieren

FOR: Profilierung kann die Komponente als verboten (forbidden) deklarieren

MUL: Profilierung kann die Komponente als m bis n mal vorkommend deklarieren

IGN: Profilierung kann die Komponente als ignoriert (ignored) deklarieren

VAL: Der Wert (value) ist profilierbar (nur Elemente mit Dateninhalt und Attribute)

2.5 Graphische Symbole

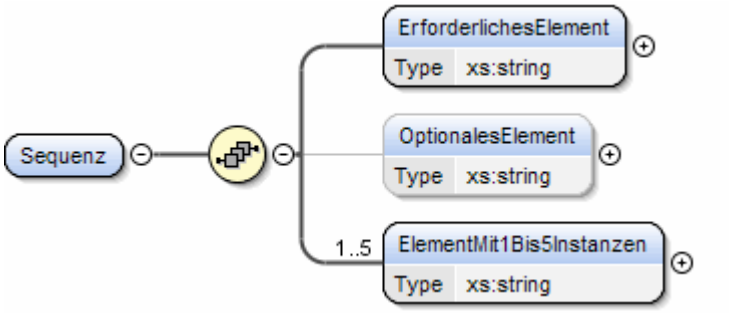
Die Strukturen und Inhalte der eXTra-Elemente werden mittels der unten beschriebenen generierten graphischen Symbole dargestellt.

Hinweis: In den meisten Fällen erscheinen Elemente und Attribute in einem im Farbdruck gelb eingefärbten Rechteck. Dieses stellt eine XSD-Typdefinition dar und trägt deren Namen.

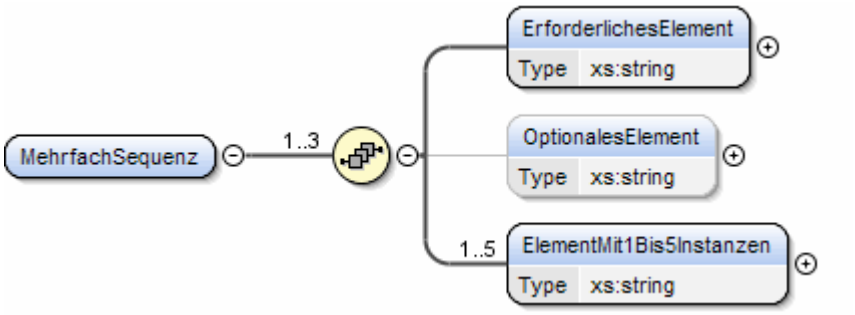
Typdefinitionen sind schemaspezifisch und daher nicht normativ. Aus diesem Grund findet sich in den folgenden Mustergraphiken auch kein Beispiel für eine Typdefinition.

Hinweis: Elemente, Sequenzen und Auswahlsequenzen, die nicht fehlen dürfen, sind mit der sie beinhaltenden Komponente mittels einer **fett** dargestellten Linie verbunden. Die Symbole von Pflichtelementen und -attributen haben einen **fetten** Rand.

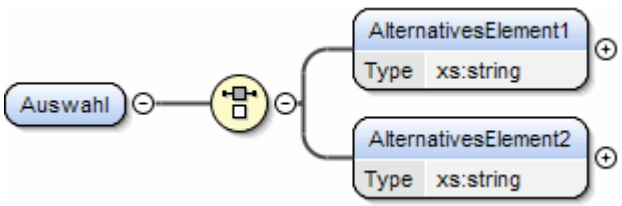
Eine Sequenz von Elementen



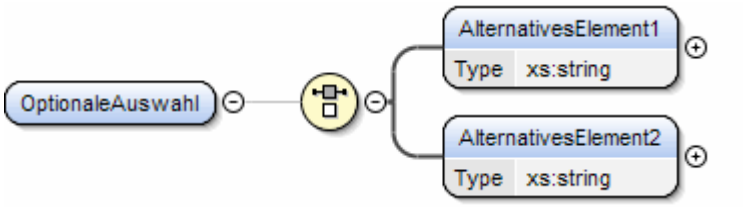
Eine bis zu dreimal wiederholte Sequenz von Elementen



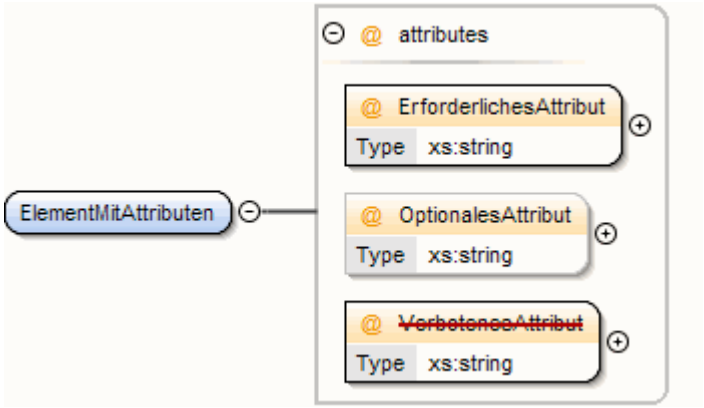
Eine erforderliche Auswahl (Kindelemente sind Pflichtelemente)



Eine optionale Auswahl (Kindelemente sind Pflichtelemente)



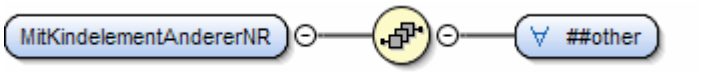
Ein Element mit Attributen



Ein Element mit einem beliebigen Kindelement aus dem eigenen Namensraum



Ein Element mit einem beliebigen Kindelement aus einem anderen Namensraum



---

3 Aufbau von Standardnachrichten [normativ]

3.1 Namensräume und Präfixe

Für eXTra-Standardnachrichten finden die nachfolgend aufgeführten Namensräume und Präfixe Anwendung. Die Festlegung und Verwendung eines einheitlichen Präfix für einen XML Namensraum ist eine Best Practice, und es wird daher EMPFOHLEN, nur die genannten Präfixe zu verwenden.

Namensraum	Präfix
http://www.extra-standard.de/namespace/message/1	xmsg

Zur Unterscheidung einzelner Standardnachrichten können ALTERNATIV die folgenden Präfixe verwendet werden (in [] stehende Werte sind obsolet und SOLLEN NICHT mehr angewendet werden):

Standardnachricht	Präfix
ConfirmationOfReceipt	xmcor [xcor]
DataRequest	xmdrq [xdrq]
ListOfConfirmationOfReceipt	xmcor [xcor]
ListOfDataRequest	xmdrq [xdrq]
ListOfStatusResponse	xmsrp
ListRequest	xmlrq
ListResponse	xmlrp
RepeatResponseRequest	xmrrr
StatusRequest	xmsrq
StatusResponse	xmsrp

4 eXTra Codelisten [normativ]

Die im Folgenden beschriebenen Codelisten sind im Schema-Modul `eXTra-messages-1.xsd` definiert.

Codelisten, die auf dem Datentyp `xsd:anyURI` basieren, haben in der Regel vordefinierte Werte aus der Domain `http://www.extra-standard.de`. In den meisten Fällen können diese verfahrensspezifisch erweitert werden. Dieser Vorgang wird als *Wertindividualisierung* bezeichnet. Ein vordefinierter Wert wird dabei durch einen *Individualisierungsausdruck* erweitert. Die Syntax eines Individualisierungsausdrucks ist abgeleitet von der Syntax sogenannter *Anker*, wie sie in URLs (die syntaktisch mit URIs identisch sind) der Adressierung von Dokumentfragmenten dienen.

Ein Individualisierungsausdruck beginnt mit einem ‚#' Zeichen. Diesem folgt eine Sequenz beliebig vieler Ziffern und/oder Buchstaben des englischen Alphabetes in Groß- oder Kleinschreibung. Diese Sequenz kann durch einzelne Minuszeichen und Punkte unterteilt werden. Vor und nach einem Minuszeichen oder Punkt MUSS eine Ziffer oder ein Buchstabe stehen. Individualisierungsausdrücke werden mit dem regulären Ausdruck

```
(#[a-zA-Z0-9]([.\-][a-zA-Z0-9]+)*)?
```

geprüft Beispiel:.

Gültige Ausdrücke		***Ungültige Ausdrücke***	
`#0`	Nur Ziffern erlaubt	`0`	‚#' fehlt
`#noSig`	Nur Buchstaben erlaubt	`#no/Sig`	Unzulässiges Zeichen ‚/'
`#A-1.B`	Gültige Unterteilung	`#.A--1`	Ungültige Unterteilung

Einige Codelisten lassen zudem die Angabe von URIs beliebiger Domains zu.

4.1 DataRequestPropertyNames

	xmsg:DataRequestPropertyNamesType
Inhalt	Menge der per *Data Request* abfragbaren Eigenschaften
Datentyp	`xsd:anyURI`
Individualisierbar	Nein
Beliebige Domain	Ja

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/RequestCreationTimeStamp
http://www.extra-standard.de/property/RequestFileName
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
http://www.extra-standard.de/property/State
```

Beschreibung

Die Codeliste definiert die Menge der per *DataRequest* abfragbaren Eigenschaften und MUSS ein Subset der Werte der Codeliste `xmsg:PropertyNamesType` darstellen.

4.2 EventNames

	xmsg:EventNamesType
Inhalt	Menge der abfragbaren Ereignisse
Datentyp	`xsd:anyURI`
Individualisierbar	Nein
Beliebige Domain	Ja

Vordefinierte Werte:

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

Beschreibung

Die Codeliste enthält URIs, die typische Ereignisse im Ablauf eines eXTra-basierten Kommunikationsszenarios darstellen.

4.3 ExtraLayerNames

Die Codeliste enumeriert die Ebenen der eXTra-Kommunikationsarchitektur.

	`xmsg:ExtraLayerNamesType`
Inhalt	Ebenen der eXTra Kommunikationsarchitektur
Datentyp	`xsd:anyURI`
Individualisierbar	Nein
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
```

4.4 ListRequestPropertyNames

Die Codeliste definiert die Menge der per *ListRequest* abfragbaren Eigenschaften und MUSS ein Subset der Werte der Codeliste `xmsg:PropertyNamesType` darstellen.

	`xmsg:ListRequestPropertyNamesType`
Inhalt	Menge der per *List Request* abfragbaren Eigenschaften
Datentyp	`xsd:anyURI`
Individualisierbar	Nein
Beliebige Domain	Ja

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/RequestCreationTimeStamp
http://www.extra-standard.de/property/RequestFileName
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
http://www.extra-standard.de/property/State
```

4.5 ListResponsePropertyNames

Die Codeliste definiert die Menge der per *ListResponse* beschreibbaren Eigenschaften und MUSS ein Subset der Werte der Codeliste `xmsg:PropertyNamesType` darstellen.

	`xmsg:ListResponsePropertyNamesType`
Inhalt	Menge der per *List Response* beschreibbaren Eigenschaften
Datentyp	`xsd:anyURI`
Individualisierbar	Nein
Beliebige Domain	Ja

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/PayloadSize
http://www.extra-standard.de/property/RequestID
http://www.extra-standard.de/property/RequestCreationTimeStamp
http://www.extra-standard.de/property/RequestFileName
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
http://www.extra-standard.de/property/State
```

4.6 PropertyNames

Die Codeliste enthält die Obermenge aller URIs, die abfragbare Eigenschaften von eXTra-Nachrichten repräsentieren.

	xmsg: PropertyNamesType
Inhalt	Menge allgemein abfragbarer Eigenschaften
Datentyp	`xsd:anyURI`
Individualisierbar	Nein
Beliebige Domain	Ja

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/PayloadSize
http://www.extra-standard.de/property/RequestID
http://www.extra-standard.de/property/RequestCreationTimeStamp
http://www.extra-standard.de/property/RequestFileName
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
http://www.extra-standard.de/property/State
```

4.7 RequestState

	xmsg:RequestStatesType
Inhalt	Stati fachlicher Nachrichten, die der eXTra-DeliveryServer auf Grund einer DataRequest Anforderung ausliefern kann
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Ja

Vordefinierte Werte:

```
http://www.extra-standard.de/requestState/NONE
http://www.extra-standard.de/requestState/AVAILABLE
http://www.extra-standard.de/requestState/FETCHED
http://www.extra-standard.de/requestState/CONFIRMED
```

Beschreibung

Die Codeliste definiert die Stati, die die fachlichen Nachrichten haben können, die der eXTra-DeliveryServer auf Grund einer DataRequest Anforderung ausliefern kann.

4.8 SortingOrder

	xmsg:SortingOrderType
Inhalt	Sortierung
Datentyp	`xsd:string`
Individualisierbar	Nein
Beliebige Domain	Ja

Vordefinierte Werte:

```
A
ASC
ASCENDING
D
DESC
DESCENDING
```

Beschreibung

Die Codeliste definiert die Art der Sortierung, in die die angeforderten auszuliefernden Elemente gebracht werden sollen.

4.9 StatusNames

Die Codeliste definiert die Stati, die ein eXTra-Request an spezifischen Kontrollpunkten (Checkpoints) der Prozesskette haben kann.

	xmsg:StatusNamesType
Inhalt	Stati eines eXTra-Request
Datentyp	`xsd:anyURI`
Individualisierbar	Nein
Beliebige Domain	Ja

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
```

4.10 StatusRequestPropertyNames

Die Codeliste definiert die Menge der per *Status Request* abfragbaren Eigenschaften und MUSS ein Subset der Werte der Codeliste `xmsg:PropertyNamesType` darstellen.

	xmsg:StatusRequestPropertyNamesType
Inhalt	Menge der per *Status Request* abfragbaren Eigenschaften
Datentyp	`xsd:anyURI`
Individualisierbar	Nein
Beliebige Domain	Ja

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/RequestCreationTimeStamp
http://www.extra-standard.de/property/RequestFileName
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
```

4.11 WeightCodes

Die Codeliste definiert die in Flags verwendbaren Fehlergewichte.

	xmsg:WeightCodeType
Inhalt	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
```

4.12 XSDTypeCodes

Die Codeliste umfasst die verwendbaren XSD Schema Typreferenzen. Sie ist weder individualisierbar noch erweiterbar.

	xmsg:XSDPrefixedTypeCodes
Inhalt	XSD Typnamen
Datentyp	`xsd:QName`
Individualisierbar	Nein
Beliebige Domain	Nein

Vordefinierte Werte:

```
<!-- Strings and names -->

<xs:enumeration value="xs:string" />
<xs:enumeration value="xs:normalizedString" />
<xs:enumeration value="xs:token" />
<xs:enumeration value="xs:Name" />
<xs:enumeration value="xs:NCName" />
<xs:enumeration value="xs:QName" />
<xs:enumeration value="xs:language" />

<!-- Numeric -->

<xs:enumeration value="xs:float" />
<xs:enumeration value="xs:double" />
<xs:enumeration value="xs:decimal" />
<xs:enumeration value="xs:integer" />
<xs:enumeration value="xs:long" />
<xs:enumeration value="xs:int" />
<xs:enumeration value="xs:short" />
<xs:enumeration value="xs:byte" />
<xs:enumeration value="xs:positiveInteger" />
<xs:enumeration value="xs:nonPositiveInteger" />
<xs:enumeration value="xs:negativeInteger" />
<xs:enumeration value="xs:nonNegativeInteger" />
<xs:enumeration value="xs:unsignedLong" />
<xs:enumeration value="xs:unsignedInt" />
<xs:enumeration value="xs:unsignedShort" />
<xs:enumeration value="xs:unsignedByte" />

<!-- Date and time -->

<xs:enumeration value="xs:duration" />
<xs:enumeration value="xs:dateTime" />
<xs:enumeration value="xs:date" />
<xs:enumeration value="xs:time" />
<xs:enumeration value="xs:gYear" />
<xs:enumeration value="xs:gMonth" />
<xs:enumeration value="xs:gYearMonth" />
```

```
<xs:enumeration value="xs:gMonthDay" />
<xs:enumeration value="xs:gDay" />

<!-- Legacy types -->

<xs:enumeration value="xs:ID" />
<xs:enumeration value="xs:IDREF" />
<xs:enumeration value="xs:IDREFS" />
<xs:enumeration value="xs:ENTITY" />
<xs:enumeration value="xs:ENITIES" />
<xs:enumeration value="xs:NMTOKEN" />
<xs:enumeration value="xs:NMTOKENS" />
<xs:enumeration value="xs:NOTATION" />

<!-- Other -->

<xs:enumeration value="xs:boolean" />
<xs:enumeration value="xs:hexBinary" />
<xs:enumeration value="xs:base64Binary" />
<xs:enumeration value="xs:anyURI" />

<!-- Undefined -->

<xs:enumeration value="xmsg:UNDEFINED" />
```

5 Nachricht ConfirmationOfReceipt [normativ]

Eine Nachricht vom Typ *ConfirmationOfReceipt* ist eine Empfangsbestätigung für eine Menge von Anwendungsnachrichten.

Der Name des Wurzelelementes ist `ConfirmationOfReceipt`.

Zu Namensraum und Präfix siehe Abschnitt *Namensräume und Präfixe*.

5.1 Beispieldokument

Der folgende *ConfirmationOfReceipt* bestätigt den Empfang eines Dokumentes mit den angegebenen Eigenschaften des eXTra-Kommunikationsprotokolls.

```
<?xml                                version="1.0"                                encoding="UTF-8"?>
<xmsg:ConfirmationOfReceipt
 xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 version="1.3"
 >

    <xmsg:Property   name="http://www.extra-standard.de/property/SenderID"   type="xs:string">
        <xmsg:Value>77700000</xmsg:Value>
    </xmsg:Property>

    <xmsg:Property  name="http://www.extra-standard.de/property/ReceiverID"  type="xs:string">
        <xmsg:Value>55550000</xmsg:Value>
    </xmsg:Property>

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

    <xmsg:Property   name="http://www.extra-standard.de/property/DataType"   type="xs:anyURI">
        <xmsg:Value>http://www.extra-standard.de/datatypes/RSVNr</xmsg:Value>
    </xmsg:Property>

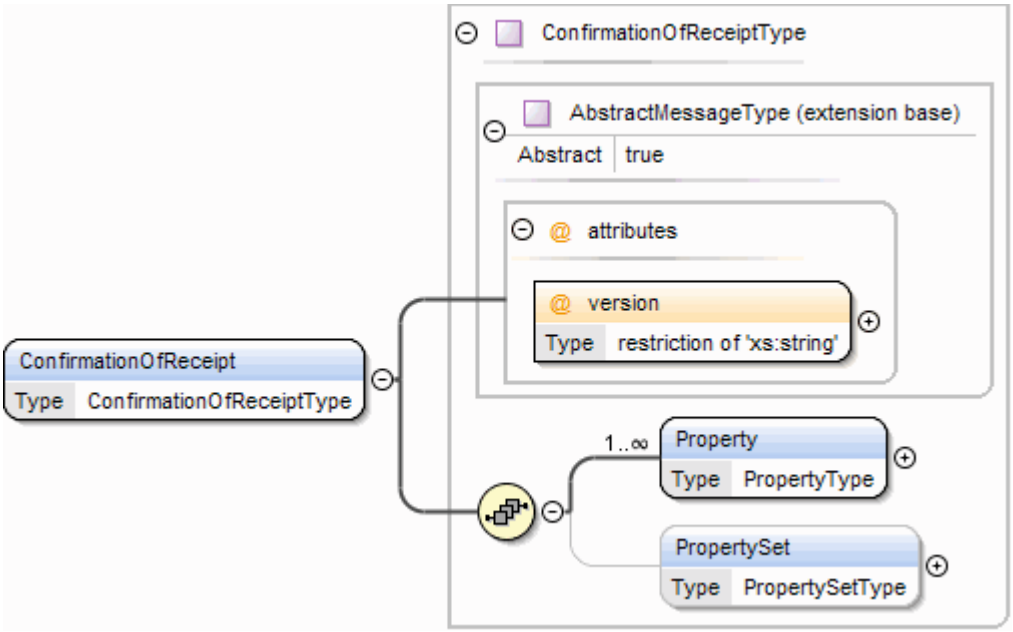
    <xmsg:Property
        name="http://www.extra-standard.de/property/ResponseCreationTimeStamp"
        type="xs:dateTime">
        <xmsg:Value>2006-05-04T18:13:51.0Z</xmsg:Value>
    </xmsg:Property>

    <xmsg:PropertySet name="http://www.extra-standard.de/property/ResponseID"
        type="xs:string"              event="http://www.extra-standard.de/event/RequestData">
        <xmsg:Value>res20090223152540-00008060</xmsg:Value>
        <xmsg:Value>res20090223152540-00008061</xmsg:Value>
        <xmsg:Value>res20090223152540-00008062</xmsg:Value>
        <xmsg:Value>res20090223152540-00008063</xmsg:Value>
        <xmsg:Value>res20090223152540-00008064</xmsg:Value>
    </xmsg:PropertySet>


</xmsg:ConfirmationOfReceipt>
```

5.2 Element ConfirmationOfReceipt

	`xmsg:ConfirmationOfReceipt`
Inhalt	Beschreibt den Empfang einer Nachricht oder eine Menge von Nachrichten.
Datentyp	`##elements`
Benutzt von	-



Attribute

	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung dieser eXTra-Standardnachricht, derzeit 1.3	
Datentyp	`xsd:string`	

Beschreibung

`xmsg:ConfirmationOfReceipt` ist das Wurzelelement einer eXTra-Nachricht vom Typ ConfirmationOfReceipt. Das Attribut `version` enthält die Versionsbezeichnung.

Eine Nachricht dieses Typs dient zur Bestätigung des Empfangs einzelner oder Mengen von *Anwendungsnachrichten*. Eine Anwendungsnachricht ist eine Nachricht in Sinne einer fachlichen Anwendung, die mittels des eXTra-Kommunikationsprotokolls übertragen wurde. Die bestätigten Anwendungsnachrichten werden mittels der Elemente `Property` und `PropertySet` anhand einer Menge von Eigenschaften beschrieben. Diese Eigenschaften sollen jeweils eine individuelle Anwendungsnachricht identifizieren, d.h. die Kombination der Werte dieser

Eigenschaften soll eindeutig sein. Für eine ausgewählte Eigenschaft erlaubt das Element `PropertySet`, mehr als einen Wert anzugeben und damit eine Menge von Anwendungsnachrichten zu identifizieren.

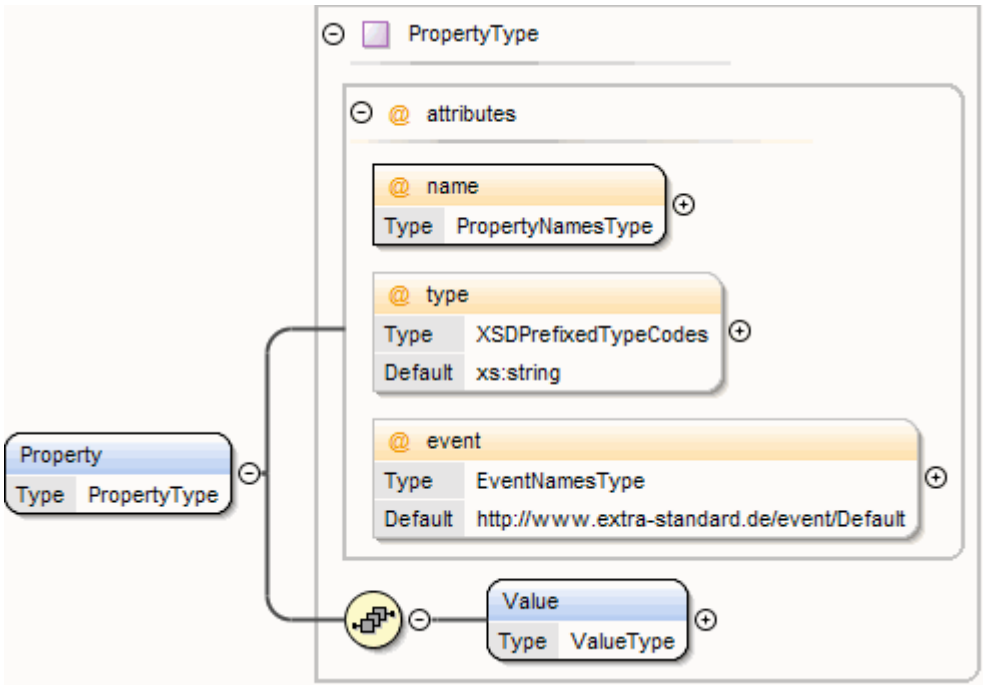
Enthält die Nachricht ConfirmationOfReceipt nur `Property`-Elemente, so wird mit dieser Folge genau eine Anwendungsnachricht identifiziert. Die `Property`-Elemente werden somit wie ein logisches UND miteinander verknüpft.

Enthält die Nachricht ConfirmationOfReceipt mehrere `Property`-Elemente und ein `PropertySet`-Element, so müssen alle Anwendungsnachrichten, die im `PropertySet`-Element aufgeführt werden allen Eigenschaften genügen, die in den `Property`-Elementen spezifiziert wurden. Die Folge von `Property`-Elementen wirken wie ein logisches UND, während die Werte im `PropertySet`-Element als logisches ODER zu verstehen sind.

Eigenschaften werden mittels einer URI identifiziert. Für Eigenschaften, die im eXTra-Kommunikationsprotokoll abgebildet werden, stehen vordefinierte URIs zur Verfügung.

5.3 Element Property

	xmsg:Property
Inhalt	Liefert für eine Eigenschaft genau einen Wert.
Datentyp	`##elements`
Benutzt von	`xmsg:ConfirmationOfReceipt`



Attribute

	@event	*optional*
Inhalt	Bezieht die abgefragte Eigenschaft auf ein Ereignis im Kommunikationsablauf	
Datentyp	`xsd:anyURI`	
Werteraum	`##codelist:EventNames`	
Default	`http:www.extra-standard.de/event/Default`	

	@name	*erforderlich*
Inhalt	Der Name der Eigenschaft	
Werteraum	`##codelist:PropertyNames`	
Datentyp	`xsd:anyURI`	

	@type	*optional*
Inhalt	Der Datentyp der Eigenschaft	
Datentyp	`xsd:string`	
Werteraum	`##codelist:XSDTypeCodes`	
Default	`xsd:string`	

Profilierung

Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`@event`						+

Beschreibung

`xmsg:Property` beschreibt eine Eigenschaft und liefert genau einen Wert dieser Eigenschaft.

Mit dem Attribut `event` wird das Ereignis im Kommunikationsablauf spezifiziert, für das der Wert der Eigenschaft gültig ist. Die vordefinierte Werteliste kann durch nutzerspezifische URIs ersetzt bzw. ergänzt werden.

Beispiel

```
<?xml                    version="1.0"                    encoding="UTF-8"?>
<xmsg:ConfirmationOfReceipt
 xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 version="1.3"
 >
    <xmsg:Property                                  name="http://www.extra-
standard.de/property/SenderID"
        type="xs:string">
        <xmsg:Value>77700000</xmsg:Value>
    </xmsg:Property>

    <xmsg:Property                                  name="http://www.extra-
standard.de/property/ReceiverID"
        type="xs:string">
        <xmsg:Value>55550000</xmsg:Value>
    </xmsg:Property>

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

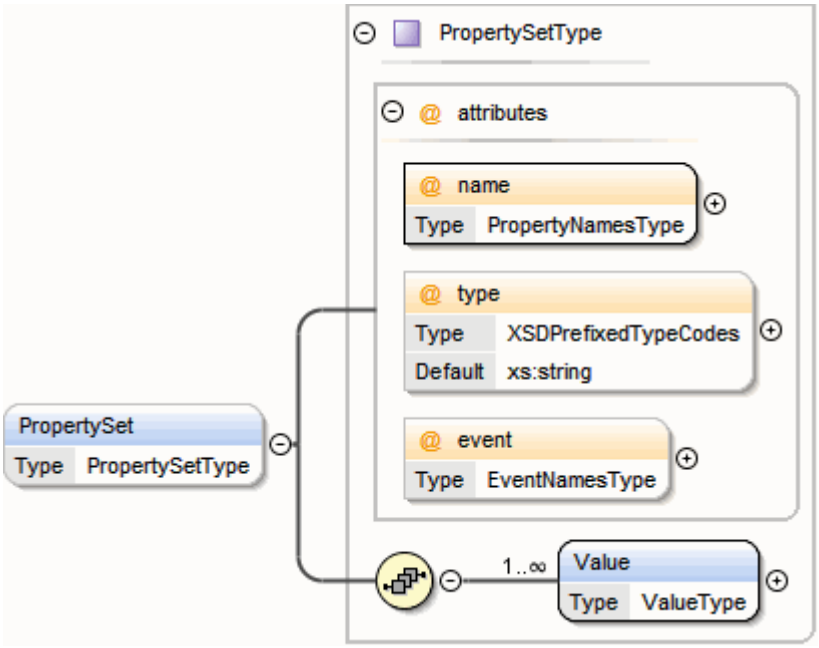
```
        /xmsg:Value>
    </xmsg:Property>

    <xmsg:Property                                    name="http://www.extra-
standard.de/property/ResponseID"
event="http://www.extra-standard.de/event/RequestData"
        <xmsg:Value>00008060</xmsg:Value>
    </xmsg:Property>

</xmsg:ConfirmationOfReceipt>
```

5.4 Element PropertySet

	xmsg:PropertySet
Inhalt	Liefert für eine Eigenschaft eine Menge von ein oder mehr Werten.
Datentyp	`##elements`
Benutzt von	`xmsg:ConfirmationOfReceipt`



Attribute

	@event	*optional*
Inhalt	Bezieht die abgefragte Eigenschaft auf ein Ereignis im Kommunikationsablauf	
Datentyp	`xsd:anyURI`	
Werteraum	`##codelist:EventNames`	
Default	`http:www.extra-standard.de/event/Default`	

	@name	*erforderlich*
Inhalt	Der Name der Eigenschaft	
Werteraum	`##codelist:PropertyNames`	
Datentyp	`xsd:anyURI`	

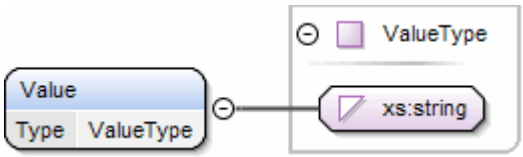
	@type	*optional*
Inhalt	Der Datentyp der Eigenschaft	
Werteraum	`##codelist:XSDTypeCodes`	
Datentyp	`xsd:string`	

Beschreibung

`xmsg:PropertySet` beschreibt eine Eigenschaft und liefert eine beliebig große Menge von Werten dieser Eigenschaft.

5.5 Element Value

	`xmsg:Value`
Inhalt	Enthält den Wert einer Eigenschaft.
Datentyp	`xsd:string`
Benutzt von	`xmsg:Property`, `xmsg:PropertySet`



Beschreibung

Enthält den Wert einer Eigenschaft. Der Datentyp wird durch das Attribut `type` des übergeordneten `Property` oder `PropertySet` Elementes festgelegt.

6 Nachricht DataRequest [normativ]

Mit Nachrichten vom Typ *DataRequest* ist es möglich, Anwendungsnachrichten oder Daten anhand bestimmter Kriterien abzurufen.

Der Name des Wurzelelementes ist `DataRequest`.

Zu Namensraum und Präfix siehe Abschnitt *Namensräume und Präfixe*.

6.1 Beispieldokument

Der folgende *DataRequest* fragt eine Reihe von Eigenschaften des eXTra-Kommunikationsprotokolls ab und begrenzt die Größe der Antwort.

```
<?xml                                version="1.0"                                encoding="UTF-8"?>
<xmsg:DataRequest
    xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    version="1.3"
    >

    <xmsg:Query>

        <xmsg:Argument property="http://www.extra-standard.de/property/SenderID"
            type="xs:string">
            <xmsg:EQ>5550000</xmsg:EQ>
        </xmsg:Argument>

        <xmsg:Argument property="http://www.extra-standard.de/property/ReceiverID"
            type="xs:string">
            <xmsg:EQ>7770000</xmsg:EQ>
        </xmsg:Argument>

        <xmsg:Argument property="http://www.extra-standard.de/property/Procedure"
            type="xs:anyURI">
            <xmsg:EQ>http://my.procedure.com/proc1</xmsg:EQ>
        </xmsg:Argument>

        <xmsg:Argument property="http://www.extra-standard.de/property/DataType"
            type="xs:anyURI">
            <xmsg:EQ>http://my.datatype.com/type1</xmsg:EQ>
        </xmsg:Argument>

        <xmsg:Argument property="http://www.extra-standard.de/property/ResponseID"
            type="xs:string"
            order="ASCENDING"
            event="http://www.extra-standard.de/event/RequestData">
            <xmsg:IN>
                <xmsg:EQ>R46934503</xmsg:EQ>
                <xmsg:EQ>R46934504</xmsg:EQ>
                <xmsg:EQ>R46934508</xmsg:EQ>
            </xmsg:IN>
        </xmsg:Argument>

        <xmsg:Argument property="http://www.extra-standard.de/property/State"
            type="xs:string"
```

```
            <xmsg:EQ>http://www.extra-standard.de/requeststate/AVAILABLE            </xmsg:EQ>
        </xmsg:Argument>

        <xmsg:Argument
            property="http://www.extra-standard.de/property/ResponseCreationTimeStamp"
            order=”ASC”                                                    type="xs:dateTime">
            <xmsg:GT>2013-10-16T09:00:00</xmsg:GT>
            <xmsg:LE>2013-10-17T09:00:00</xmsg:LE>
        </xmsg:Argument>

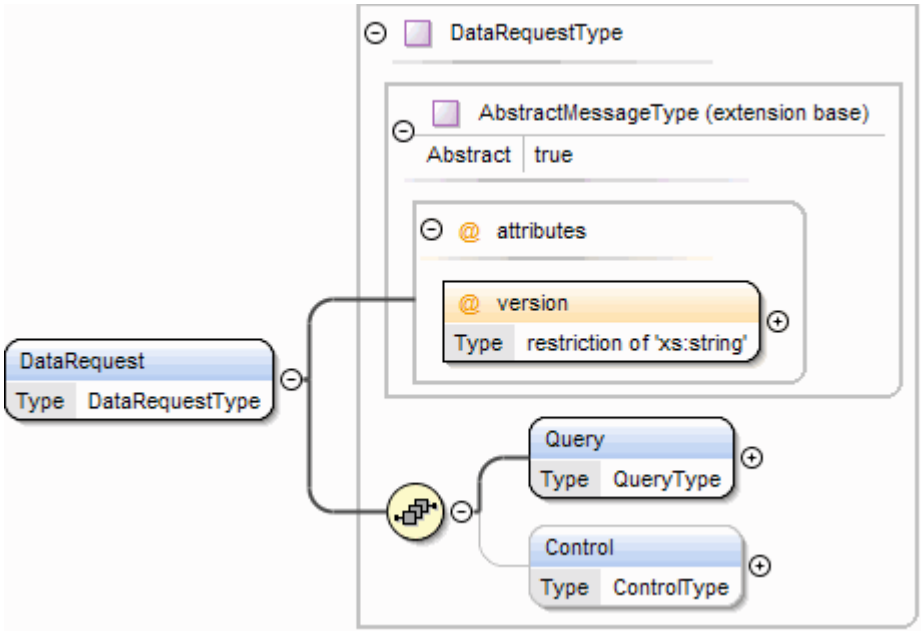
    </xmsg:Query>

    <xmsg:Control>
        <xmsg:MaximumPackages>10</xmsg:MaximumPackages>
        <xmsg:MaximumMessages>100</xmsg:MaximumMessages>
        <xmsg:MaximumSize>10M</xmsg:MaximumSize>
    </xmsg:Control>

</xmsg:DataRequest>
```

6.2 Element DataRequest

	`xmsg:DataRequest`
Inhalt	Beschreibt eine Anforderung von Anwendungsnachrichten oder Daten anhand bestimmter Kriterien.
Datentyp	`##elements`
Benutzt von	-



Attribute

	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung dieser eXTra-Standardnachricht, derzeit 1.3	
Datentyp	`xsd:string`	

Beschreibung

Eine Nachricht dieses Typs dient zur Anforderung einzelner oder Mengen von Daten oder *Anwendungsnachrichten*. Eine Anwendungsnachricht ist eine Nachricht in Sinne einer fachlichen Anwendung, die mit Hilfe des eXTra-Kommunikationsprotokolls angefordert wird. Die Kriterien der Anforderung werden mit dem Element `Query` und einer oder mehrerer Unterelemente `Argument` beschrieben. Die Kriterien der Anforderung müssen nicht notwendigerweise genau eine individuelle Anwendungsnachricht identifizieren, sie können auch für eine Menge von bereitgestellten Anwendungsnachrichten zutreffen. Um für diesen Fall die Menge der zurück zu liefernden Anwendungsnachrichten bzw. die Größe des zurück zu lie-

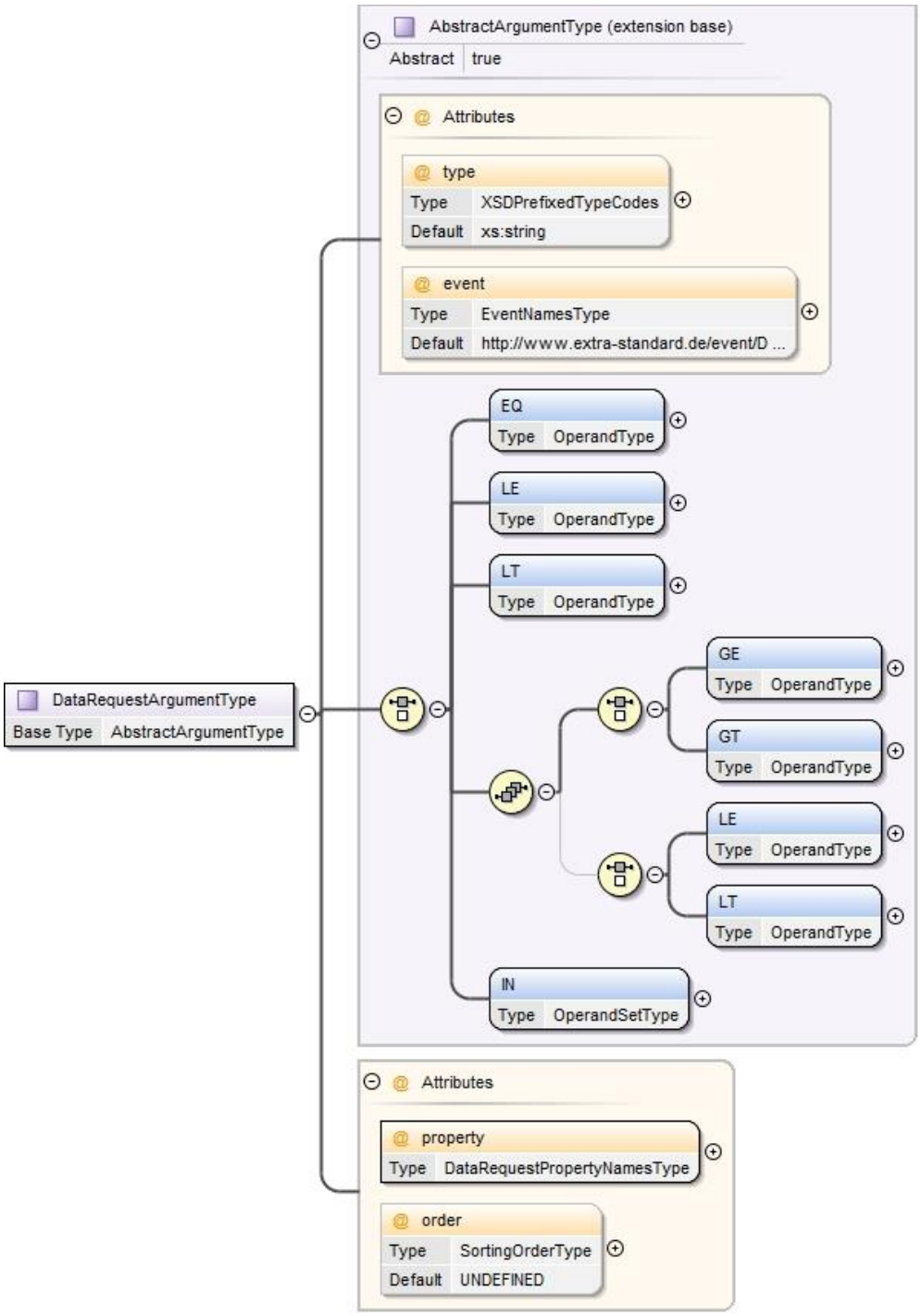
fernden eXTra-Dokumentes (genauer des Elements `TransportBody`) begrenzen zu können, gibt es das Element `Control`.

Enthält die Query mehrere `Argument`-Elemente, so müssen die angeforderten Anwendungsnachrichten bzw. Daten allen so spezifizierten Kriterien genügen. Die einzelnen Kriterien werden somit wie ein logisches UND miteinander verknüpft.

Ein Kriterium besteht aus einer abgefragten Eigenschaft (Attribut `property`), die per URI referenziert wird, und einem Wert, mit den die entsprechende Eigenschaft der abgefragten Objekte verglichen wird. Der Wert wird als Einzelwert mit einem Kindelement oder als Wertebereich mit inklusiver Unter- und Obergrenze mit zwei Kindelementen angegeben. Ein Kindelement bestimmt zugleich den Operator für den Vergleich mit Einzelwert bzw. der Unter- oder Obergrenze.

6.3 Element Argument

	xmsg:Argument
Inhalt	Beschreibt ein Kriterium einer Abfrage
Datentyp	`##elements`
Benutzt von	`xmsg:Query`



Attribute

	@event	*optional*
Inhalt	Bezieht die abgefragte Eigenschaft auf ein Ereignis im Kommunikationsablauf	
Datentyp	`xsd:anyURI`	
Werteraum	`##codelist:EventNames`	
Default	`http:www.extra-standard.de/event/Default`	

	@order	*optional*
Inhalt	Bezeichnet die gewünschte Sortierung der auszuliefernden Anwendungsnachrichten bzw. Daten	
Werteraum	`##codelist:SortingOrder`	
Datentyp	`xsd:string`	

	@property	*erforderlich*
Inhalt	Bezeichnet die abgefragte Eigenschaft	
Werteraum	`##codelist:DataRequestPropertyNames`	
Datentyp	`xsd:anyURI`	

	@type	*optional*
Inhalt	Der XSD Datentyp der abgefragten Eigenschaft	
Datentyp	`xsd:string`	
Werteraum	`##codelist:XSDTypeCodes`	
Default	`xsd:string`	

Profilierung

Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
@event						+

Beschreibung

Ein Kriterium besteht aus einer abgefragten Eigenschaft (Attribut `property`), die per URI referenziert wird, und einem Werteraum, mit dem die Werte der abgefragten Objekte verglichen werden. Der Werteraum wird als Einzelwert oder als Wertebereich mit inklusiver Unter- und Obergrenze oder als Werteliste angegeben. Die wertetragenden Elemente des Werteraums bestimmen zugleich den Operator für den Vergleich mit Einzelwerten bzw. der Unter- oder Obergrenze.

Das Attribut `type` spezifiziert den Datentyp des Arguments.

Mit dem Attribut `event` wird das Ereignis im Kommunikationsablauf spezifiziert, für das die Eigenschaft abgefragt werden soll. Die vordefinierte Werteliste kann durch nutzerspezifische URIs ersetzt bzw. ergänzt werden.

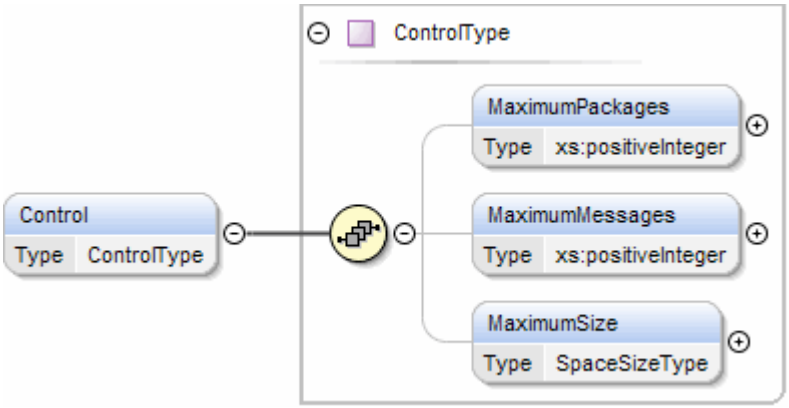
Die Reihenfolge der auszuliefernden Anwendungsnachrichten bzw. Daten kann mit dem Attribut `order` festgelegt werden.

Beispiel

```
<xmsg:Argument property="http://www.extra-standard.de/property/ResponseCreationTimeStamp"
    order="ASC"
    type="xs:dateTime">
    <xmsg:GT>2009-03-16T09:00:00</xmsg:GT>
    <xmsg:LE>2009-03-17T09:00:00</xmsg:LE>
</xmsg:Argument>
```

6.4 Element Control

	xmsg:Control
Inhalt	Enthält Angaben zur Steuerung der Erzeugung der Antwort auf die Datenanforderung.
Datentyp	`##elements`
Benutzt von	`xmsg:DataRequest`



Beschreibung

Enthält Angaben zur Steuerung der Erzeugung der Antwort auf die Datenanforderung. Die Größe der Antwort kann durch mehrere optionale Parameter beschränkt werden. Ist mehr als ein Parameter angegeben, wirkt der, dessen Beschränkung sich als stärkste erweist.

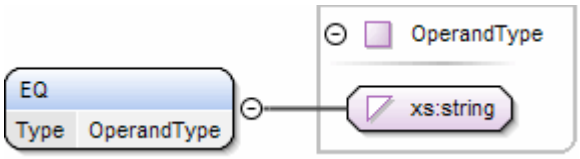
Mit dem Element `MaximumSize` wird die maximale Größe der Antwort auf die Datenanforderung im Element `TransportBody` festgelegt.

Beispiel

```
<xmsg:Control>
    <xmsg:MaximumMessages>100</xmsg:MaximumMessages>
</xmsg:Control>
```

6.5 Element EQ

	`xmsg:EQ`
Inhalt	Der Wert eines Kriteriums
Datentyp	`xsd:string`
Benutzt von	`xmsg:Argument`

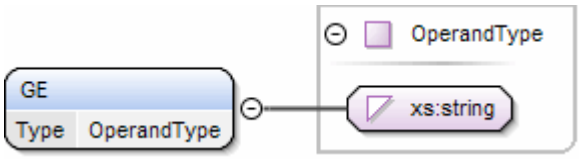


Beschreibung

Enthält den Wert eines Kriteriums. Der Vergleich soll gemäß dem in `xsd:Argument/@type` spezifizierten Typ typgerecht erfolgen. Ein Wert erfüllt den Vergleich, wenn er als GLEICH dem Wert des Kriteriums gilt.

6.6 Element GE

	xmsg:GE
Inhalt	Der Wert eines Kriteriums
Datentyp	`xsd:string`
Benutzt von	`xmsg:Argument`

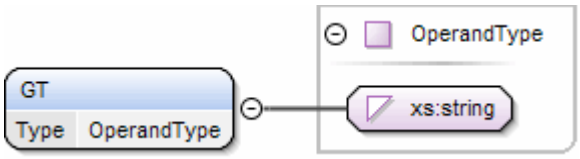


Beschreibung

Enthält den Wert eines Kriteriums. Der Vergleich soll gemäß dem in `xsd:Argument/@type` spezifizierten Typ typgerecht erfolgen. Ein Wert erfüllt den Vergleich, wenn er als GRÖSSER oder GLEICH dem Wert des Kriteriums gilt.

6.7 Element GT

	`xmsg:GT`
Inhalt	Der Wert eines Kriteriums
Datentyp	`xsd:string`
Benutzt von	`xmsg:Argument`

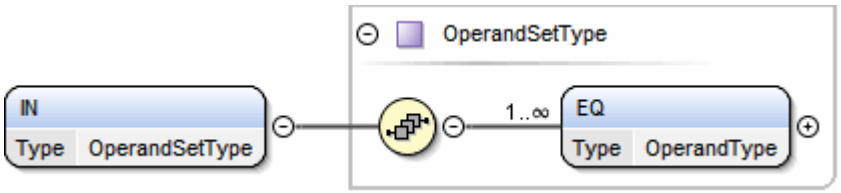


Beschreibung

Enthält den Wert eines Kriteriums. Der Vergleich soll gemäß dem in `xsd:Argument/@type` spezifizierten Typ typgerecht erfolgen. Ein Wert erfüllt den Vergleich, wenn er als GRÖSSER als der Wert des Kriteriums gilt.

6.8 Element IN

	`xmsg:IN`
Inhalt	Eine Liste von Werten eines Kriteriums
Datentyp	`xsd:string`
Benutzt von	`xmsg:Argument`



Beschreibung

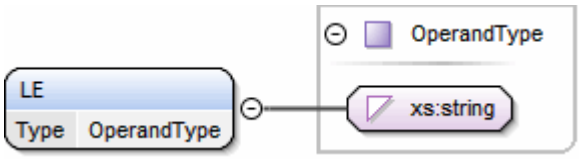
Enthält Werte eines Kriteriums. Der Vergleich soll gemäß dem in `xsd:Argument/@type` spezifizierten Typ typgerecht erfolgen. Ein Wert erfüllt den Vergleich, wenn er als GLEICH einem der Werte des Kriteriums gilt.

Beispiel

```
<xmsg:Argument property="http://www.extra-standard.de/property/ResponseID"
    type="xs:string"
    event="http://www.extra-standard.de/event/RequestData">
    <xmsg:IN>
        <xmsg:EQ>R46934503</xmsg:EQ>
        <xmsg:EQ>R46934504</xmsg:EQ>
        <xmsg:EQ>R46934508</xmsg:EQ>
    </xmsg:IN>
</xmsg:Argument>
```

6.9 Element LE

	`xmsg:LE`
Inhalt	Der Wert eines Kriteriums
Datentyp	`xsd:string`
Benutzt von	`xmsg:Argument`

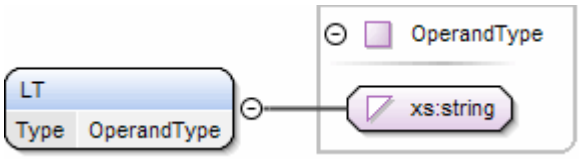


Beschreibung

Enthält den Wert eines Kriteriums. Der Vergleich soll gemäß dem in `xsd:Argument/@type` spezifizierten Typ typgerecht erfolgen. Ein Wert erfüllt den Vergleich, wenn er als KLEINER oder GLEICH dem Wert des Kriteriums gilt.

6.10 Element LT

	xmsg:LT
Inhalt	Der Wert eines Kriteriums
Datentyp	`xsd:string`
Benutzt von	`xmsg:Argument`



Beschreibung

Enthält den Wert eines Kriteriums. Der Vergleich soll gemäß dem in `xsd:Argument/@type` spezifizierten Typ typgerecht erfolgen. Ein Wert erfüllt den Vergleich, wenn als KLEINER als der Wert des Kriteriums gilt.

6.11 Element MaximumMessages

	xmsg:MaximumMessages
Inhalt	Eine positive Ganzzahl
Datentyp	`xsd:positiveInteger`
Benutzt von	`xmsg:Control`



Beschreibung

`xmsg:MaximumMessages` gibt an, wie viele Anwendungsnachrichten maximal übertragen werden sollen, um die Anforderung zu befriedigen.

Beispiel

```
<xmsg:Control>
    <xmsg:MaximumMessages>100</xmsg:MaximumMessages>
</xmsg:Control>
```

6.12 Element MaximumPackages

	xmsg:MaximumMessages
Inhalt	Eine positive Ganzzahl
Datentyp	`xsd:positiveInteger`
Benutzt von	`xmsg:Control`



Beschreibung

`xmsg:MaximumPackages` gibt an, aus wieviel Paketen eine Antwort maximal bestehen darf, um die Anforderung zu befriedigen.

Beispiel

```
<xmsg:Control>
    <xmsg:MaximumPackages>10</xmsg:MaximumPackages>
</xmsg:Control>
```

6.13 Element MaximumResults

	`xmsg:MaximumResults`
Inhalt	Eine positive Ganzzahl
Datentyp	`xsd:positiveInteger`
Benutzt von	`xmsg:Control`



Beschreibung

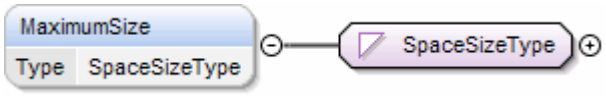
`xmsg:MaximumResults` gibt an, wieviele Treffer maximal übertragen werden sollen, um die Anforderung zu befriedigen.

Beispiel

```
<xmsg:Control>
    <xmsg:MaximumResults>32</xmsg:MaximumResults>
</xmsg:Control>
```

6.14 Element MaximumSize

	xmsg:MaximumSize
Inhalt	Eine positive Ganzzahl
Datentyp	`xmsg:SpaceSizeType`
Benutzt von	`xmsg:Control`



Beschreibung

Gibt an, wie groß die Antwort im Element `TransportBody` maximal sein darf, um die Anforderung zu befriedigen. Der Wert hat die Form *`nU`, wobei* `n` eine Ganzahl > 0 und `U` einer der folgenden Großbuchstaben ist, der die Einheit angibt:

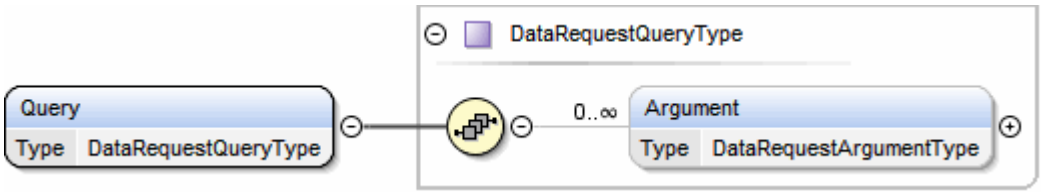
`B`	Byte
`K`	Kilobyte
`M`	Megabyte
`G`	Gigabyte

Beispiel

```
<xmsg:Control>
    <xmsg:MaximumSize>10M</xmsg:MaximumSize>
</xmsg:Control>
```

6.15 Element Query

	`xmsg:Query`
Inhalt	Beschreibt den Abfrageteil der Anforderung.
Datentyp	`##elements`
Benutzt von	`xmsg:DataRequest`



Beschreibung

Beschreibt eine Abfrage als ein Set von Kriterien (Element `Argument`). Jedes Kriterium ist mit einem Typ behaftet, spezifiziert einen oder mehrere Vergleichswerte und kann auf ein spezifisches Ereignis in einem eXTra-basierten Prozess bezogen werden.

7 Nachricht ListRequest [normativ]

Mit der Nachricht des Typs *ListRequest* wird eine Funktion zur Verfügung gestellt, die Auskunft über die verfügbaren Anwendungsnachrichten des eXTra-DeliveryServer gibt.

Der Name des Wurzelelementes ist `ListRequest`.

Zu Namensraum und Präfix siehe Abschnitt *Namensräume und Präfixe*.

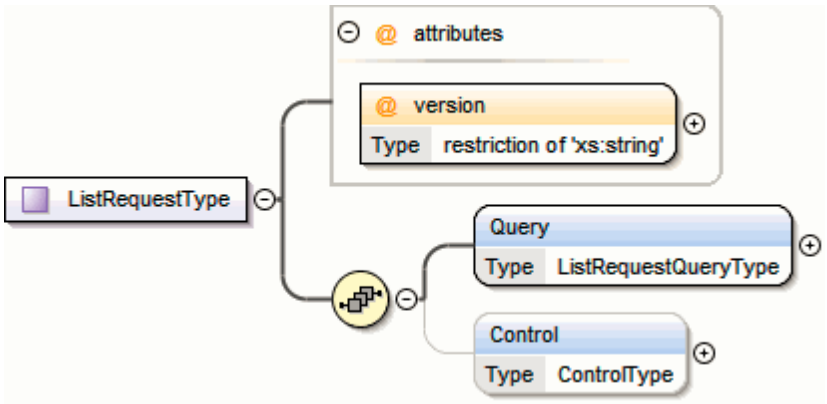
7.1 Beispieldokument

```
<?xml                              version="1.0"                              encoding="UTF-8"?>
<xmsg:ListRequest
    xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    version="1.0"
    >
    <xmsg:Query>
        <xmsg:Argument              property="http://www.extra-standard.de/property/DataType">
            <xmsg:EQ>AAAAAA</xmsg:EQ>
        </xmsg:Argument>
        <xmsg:Argument             property="http://www.extra-standard.de/property/ResponseID"
            order="ASCENDING">
            <xmsg:GT>12345</xmsg:GT>
        </xmsg:Argument>
        <xmsg:Argument                  property="http://www.extra-standard.de/property/State"
            type="xs:string">
            <xmsg:EQ>http://www.extra-standard.de/requeststate/AVAILABLE            </xmsg:EQ>
        </xmsg:Argument>
    </xmsg:Query>

</xmsg:ListRequest>
```

7.2 Element ListRequest

	`xmsg:ListRequest`
Inhalt	Eine Nachricht des Typs *ListRequest*
Datentyp	`##elements`
Benutzt von	-



Attribute

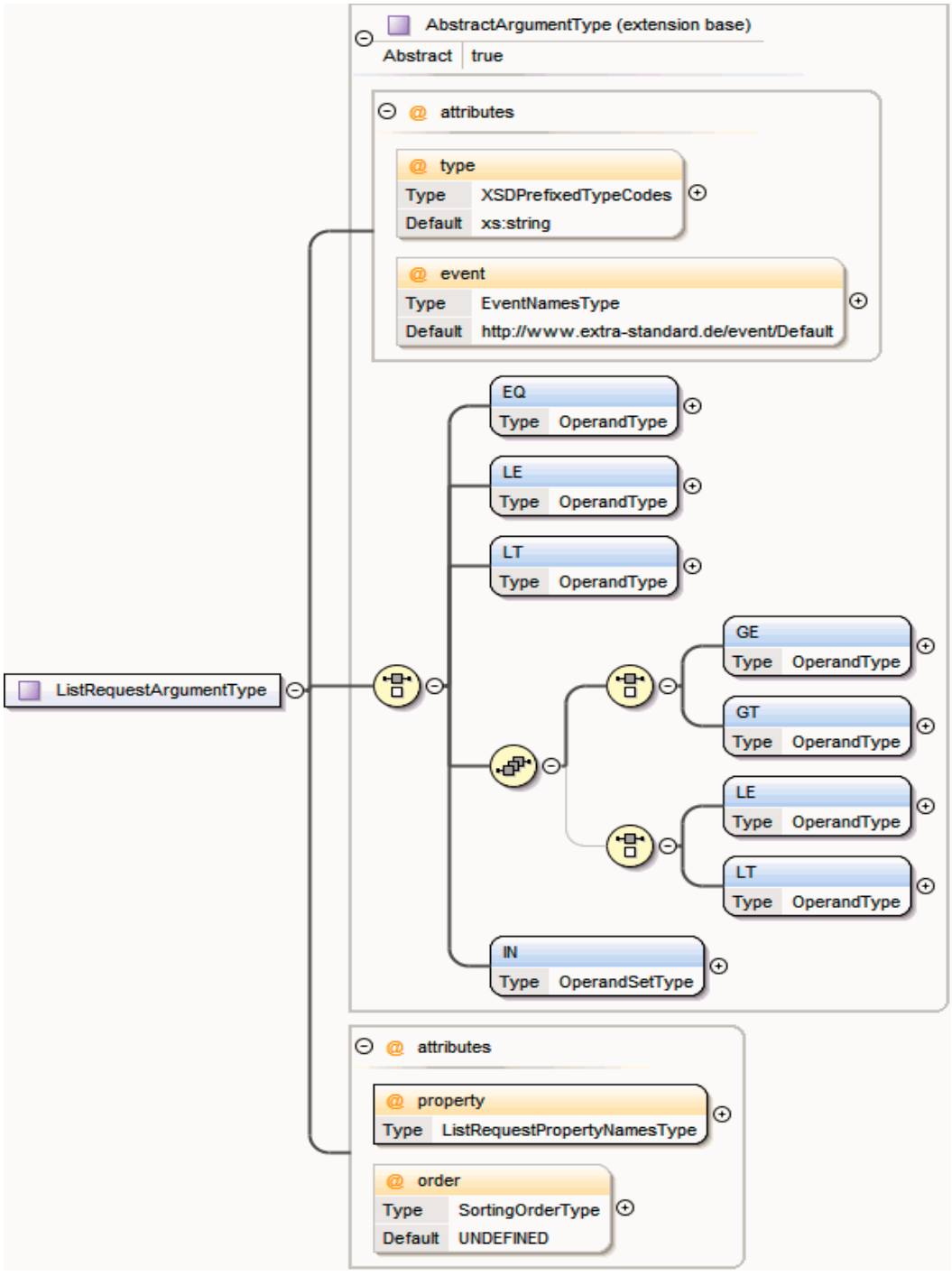
	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung dieser eXTra-Standardnachricht, derzeit 1.0	
Datentyp	`xsd:string`	

Beschreibung

`xmsg:ListRequest` ist das Wurzelelement einer eXTra-Nachricht vom Typ *ListRequest*. Das Attribut `version` enthält die Versionsbezeichnung.

7.3 Element Argument

	`xmsg:Argument`
Inhalt	Beschreibt ein Kriterium einer Abfrage
Datentyp	`##elements`
Benutzt von	`xmsg:Query`



Attribute

	@event	*optional*
Inhalt	Bezieht die abgefragte Eigenschaft auf ein Ereignis im Kommunikationsablauf	
Datentyp	`xsd:anyURI`	
Werteraum	`##codelist:EventNames`	
Default	`http:www.extra-standard.de/event/Default`	

	@order	*optional*
Inhalt	Bezeichnet die gewünschte Sortierung der Elemente der Antwort auf die Anfrage	
Werteraum	`##codelist:SortingOrder`	
Datentyp	`xsd:string`	

	@property	*erforderlich*
Inhalt	Bezeichnet die abgefragte Eigenschaft	
Werteraum	`##codelist:DataRequestPropertyNames`	
Datentyp	`xsd:anyURI`	

	@type	*optional*
Inhalt	Der XSD Datentyp der abgefragten Eigenschaft	
Datentyp	`xsd:string`	
Werteraum	`##codelist:XSDTypeCodes`	
Default	`xsd:string`	

Profilierung

Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
@event						+

Beschreibung

Ein Kriterium besteht aus einer abgefragten Eigenschaft (Attribut `property`), die per URI referenziert wird, und einem Werteraum, mit dem die Werte der abgefragten Objekte verglichen werden. Der Werteraum wird als Einzelwert oder als Wertebereich mit inklusiver Unter- und Obergrenze oder als Werteliste angegeben. Die wertetragenden Elemente des Werteraums bestimmen zugleich den Operator für den Vergleich mit Einzelwerten bzw. der Unter- oder Obergrenze.

Das Attribut `type` spezifiziert den Datentyp des Arguments.

Mit dem Attribut `event` wird das Ereignis im Kommunikationsablauf spezifiziert, für das die Eigenschaft abgefragt werden soll. Die vordefinierte Werteliste kann durch nutzerspezifische URIs ersetzt bzw. ergänzt werden.

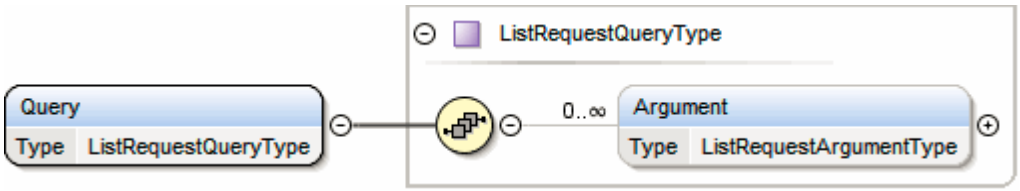
Die Sortierung der Ergebnisliste kann mit dem Attribut `order` festgelegt werden.

Beispiel

```
<xmsg:Argument property="http://www.extra-standard.de/property/ResponseID"
    type="string">
    event="http://www.extra-standard.de/event/RequestData"
    order="ASC"
    <xmsg:GT>201310140900001234</xmsg:GT>
</xmsg:Argument>
```

7.4 Element Query

	`xmsg:Query`
Inhalt	Beschreibt den Abfrageteil der Anforderung.
Datentyp	`##elements`
Benutzt von	`xmsg:StatusRequest`



Beschreibung

Beschreibt eine Abfrage als ein Set von Kriterien (Element `Argument`). Jedes Kriterium ist mit einem Typ behaftet, spezifiziert einen oder mehrere Vergleichswerte und kann auf ein spezifisches Ereignis in einem eXTra-basierten Prozess bezogen werden.

8 Nachricht ListResponse [normativ]

Mit einer Nachricht des Typs *ListResponse* wird das Ergebnis einer Auskunftsfunktion über die verfügbaren Anwendungsnachrichten im eXTra-DeliveryServer übertragen. Die Auskunftsfunktion wird typischerweise mit der Nachricht des Types *ListRequest* angefordert.

Der Name des Wurzelelementes ist `ListResponse`.

Zu Namensraum und Präfix siehe Abschnitt *Namensräume und Präfixe*.

8.1 Beispieldokument

```
<?xml version="1.0" encoding="UTF-8"?>
<xmsg:ListResponse
    xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    version="1.0"
    >

    <xmsg:Entry>
        <xmsg:Property name="http://www.extra-standard.de/property/SenderID">
            <xmsg:Value>11112222</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/ReceiverID">
            <xmsg:Value>12348765</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/Procedure">
            <xmsg:Value>DUA</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/DataType">
            <xmsg:Value>Meldung</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/ResponseID"
             event="RequestData" type="string">
            <xmsg:Value>12348765</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name=
            "http://www.extra-standard.de/property/ResponseCreationTimestamp">
            <xmsg:Value>2013-10-14:T09:08:20</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/ResponseFileName">
            <xmsg:Value>EDUA0123456</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/PayloadSize">
            <xmsg:Value>1234567B</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/State">
            <xmsg:Value>http://www.extra-standard.de/requeststate/FETCHED</xmsg:Value>
        </xmsg:Property>
    </xmsg:Entry>

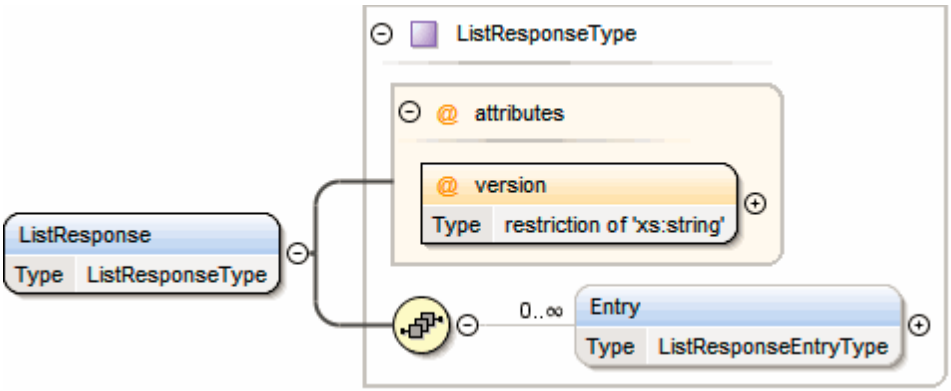
    <xmsg:Entry>
        <xmsg:Property name="http://www.extra-standard.de/property/SenderID">
            <xmsg:Value>11112222</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/ReceiverID">
            <xmsg:Value>12348765</xmsg:Value>
        </xmsg:Property>
```

```
        <xmsg:Property name="http://www.extra-standard.de/property/Procedure">
            <xmsg:Value>DUA</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/DataType">
            <xmsg:Value>Meldung</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name=
            "http://www.extra-standard.de/property/ResponseCreationTimestamp">
            <xmsg:Value>2013-10-14:T11:18:22</xmsg:Value>
    </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/ResponseFileName">
            <xmsg:Value>EDUA0123457</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/PayloadSize">
            <xmsg:Value>123444B</xmsg:Value>
        </xmsg:Property>
        <xmsg:Property name="http://www.extra-standard.de/property/State">
            <xmsg:Value>http://www.extra-standard.de/requeststate/AVAILABLE</xmsg:Value>
        </xmsg:Property>
    </xmsg:Entry>

</xmsg:ListResponse>
```

8.2 Element ListResponse

	`xmsg:ListResponse`
Inhalt	Eine Nachricht des Typs *ListResponse*
Datentyp	`##elements`
Benutzt von	-



Attribute

	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung dieser eXTra-Standardnachricht, derzeit 1.0	
Datentyp	`xsd:string`	

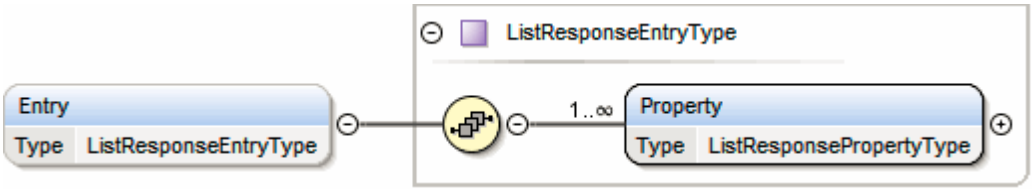
Beschreibung

`xmsg:ListResponse` ist das Wurzelelement einer eXTra-Nachricht vom Typ *ListResponse*.

Das Attribut `version` enthält die Versionsbezeichnung.

8.3 Element Entry

	`xmsg:Entry`
Inhalt	Eine Folge von Eigenschaften einer Anwendungsnachricht
Datentyp	`##elements`
Benutzt von	`xmsg:ListResponse`



Beschreibung

`xmsg:Entry` beschreibt die Eigenschaften einer Anwendungsnachricht, die der eXTra-DeliveryServer ausliefern kann.

9 Nachricht ListOfConfirmationOfReceipt [normativ]

Eine Nachricht dieses Typs dient zur Übertragung einer Menge von Nachrichten des Typs *ConfirmationOfReceipt*.

Der Name des Wurzelelementes ist `ListOfConfirmationOfReceipt`.

Zu Namensraum und Präfix siehe Abschnitt *Namensräume und Präfixe*.

9.1 Beispieldokument

```
<?xml version="1.0" encoding="UTF-8"?>
<xmsg:ListOfConfirmationOfReceipt
    xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    version="1.1"
    >

    <!-- #1 -->

    <xmsg:ConfirmationOfReceipt>
        <xmsg:Property
            name="http://www.extra-standard.de/property/SenderID"
            type="xs:string">
            <xmsg:Value>77700000</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property
            name="http://www.extra-standard.de/property/ReceiverID"
            type="xs:string">
            <xmsg:Value>55550000</xmsg:Value>
        </xmsg:Property>

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

        <xmsg:Property
            name="http://www.extra-standard.de/property/DataType"
            type="xs:anyURI">
            <xmsg:Value>http://www.extra-standard.de/datatypes/RSVNr</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property
            name="http://www.extra-standard.de/property/ResponseCreationTimeStamp"
            type="xs:dateTime">
            <xmsg:Value>2006-05-04T18:13:51.0Z</xmsg:Value>
        </xmsg:Property>

        <xmsg:Property
            name="http://www.extra-standard.de/property/ResponseID"
            type="xs:string"
            event="http://www.extra-standard.de/event/RequestData">
            <xmsg:Value>res20090223152540-00008060</xmsg:Value>
        </xmsg:Property>
```

```
    <xmsg:Property
        name="http://www.extra-standard.de/property/ResponseFileName"
        type="xs:string"
        event="http://www.extra-standard.de/event/RequestData">
        <xmsg:Value>file20090223152540-00008060</xmsg:Value>
    </xmsg:Property>

</xmsg:ConfirmationOfReceipt>

<!-- #2 -->

<xmsg:ConfirmationOfReceipt>
    <xmsg:Property
        name="http://www.extra-standard.de/property/SenderID"
        type="xs:string">
        <xmsg:Value>88800000</xmsg:Value>
    </xmsg:Property>

    <xmsg:Property
        name="http://www.extra-standard.de/property/ReceiverID"
        type="xs:string">
        <xmsg:Value>66660000</xmsg:Value>
    </xmsg:Property>

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

    <xmsg:Property
        name="http://www.extra-standard.de/property/DataType"
        type="xs:anyURI">
        <xmsg:Value>http://www.extra-standard.de/datatypes/RSVNr</xmsg:Value>
    </xmsg:Property>

    <xmsg:Property
        name="http://www.extra-standard.de/property/ResponseCreationTimeStamp"
        type="xs:dateTime">
        <xmsg:Value>2006-05-04T18:13:52.0Z</xmsg:Value>
    </xmsg:Property>

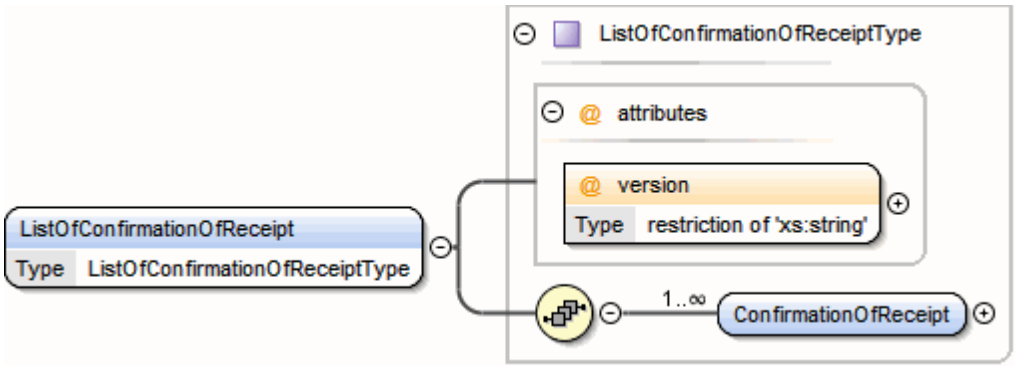
    <xmsg:Property
        name="http://www.extra-standard.de/property/ResponseID"
        type="xs:string"
        event="http://www.extra-standard.de/event/RequestData">
        <xmsg:Value>res20090223152541-00008060</xmsg:Value>
    </xmsg:Property>
```

```
        <xmsg:Property
            name="http://www.extra-standard.de/property/ResponseFileName"
            type="xs:string"
            event="http://www.extra-standard.de/event/RequestData">
            <xmsg:Value>file20090223152541-00008060</xmsg:Value>
        </xmsg:Property>

    </xmsg:ConfirmationOfReceipt>
</xmsg:ListOfConfirmationOfReceipt>
```

9.2 Element ListOfConfirmationOfReceipt

	`xmsg:ListOfConfirmationOfReceipt`
Inhalt	Eine Menge von Nachrichten des Typs *ConfirmationOfReceipt*.
Datentyp	`##elements`
Benutzt von	-



Attribute

	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung dieser eXTra-Standardnachricht, derzeit 1.1	
Datentyp	`xsd:string`	

Beschreibung

`xmsg:ListOfConfirmationOfReceipt` ist das Wurzelelement einer eXTra-Nachricht vom Typ ConfirmationOfReceipt. Das Attribut `version` enthält die Versionsbezeichnung.

Eine Nachricht dieses Typs dient zur Übertragung einer Menge von Nachrichten des Typs *ConfirmationOfReceipt*

10 Nachricht ListOfDataRequest [normativ]

Eine Nachricht dieses Typs dient zur Übertragung einer Menge von Nachrichten des Typs DataRequest.

Der lokale Name des Wurzelelementes ist `ListOfDataRequest`.

Zu Namensraum und Präfix siehe Abschnitt *Namensräume und Präfixe*.

10.1 Beispieldokument

.

```
<?xml version="1.0" encoding="UTF-8"?>
<xmsg:ListOfDataRequest
    xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    version="1.1"
    >

    <!-- #1 -->

    <xmsg:DataRequest>
        <xmsg:Query>

            <xmsg:Argument property="http://www.extra-standard.de/property/SenderID"
                type="xs:string">
                <xmsg:EQ>5550000</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument property="http://www.extra-standard.de/property/ReceiverID"
                type="xs:string">
                <xmsg:EQ>7770000</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument property="http://www.extra-standard.de/property/Procedure"
                type="xs:anyURI">
                <xmsg:EQ>http://my.procedure.com/proc1</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument property="http://www.extra-standard.de/property/DataType"
                type="xs:anyURI">
                <xmsg:EQ>http://my.datatype.com/type1</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument property="http://www.extra-standard.de/property/ResponseID"
                type="xs:string"
                event="http://www.extra-standard.de/event/RequestData">
                <xmsg:EQ>R46934503</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument
                property="http://www.extra-standard.de/property/ResponseFileName"
                type="xs:string"
                event="http://www.extra-standard.de/event/RequestData">
                <xmsg:EQ>R46934503</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument
                property="http://www.extra-standard.de/property/ResponseCreationTimeStamp"
                type="xs:dateTime">
                <xmsg:GT>2009-03-16T09:00:00</xmsg:GT>
                <xmsg:LE>2009-03-17T09:00:00</xmsg:LE>
            </xmsg:Argument>

        </xmsg:Query>

        <xmsg:Control>
            <xmsg:MaximumPackages>10</xmsg:MaximumPackages>
            <xmsg:MaximumMessages>100</xmsg:MaximumMessages>
            <xmsg:MaximumSize>10M</xmsg:MaximumSize>
        </xmsg:Control>

    </xmsg:DataRequest>

    <!-- #2 -->

    <xmsg:DataRequest>
        <xmsg:Query>

            <xmsg:Argument property="http://www.extra-standard.de/property/SenderID"
                type="xs:string">
                <xmsg:EQ>5550000</xmsg:EQ>
```

```
            </xmsg:Argument>

            <xmsg:Argument property="http://www.extra-standard.de/property/ReceiverID"
                type="xs:string">
                <xmsg:EQ>7770000</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument property="http://www.extra-standard.de/property/Procedure"
                type="xs:anyURI">
                <xmsg:EQ>http://my.procedure.com/proc1</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument property="http://www.extra-standard.de/property/DataType"
                type="xs:anyURI">
                <xmsg:EQ>http://my.datatype.com/type1</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument property="http://www.extra-standard.de/property/ResponseID"
                type="xs:string"
                event="http://www.extra-standard.de/event/RequestData">
                <xmsg:EQ>R46934503</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument
                property="http://www.extra-standard.de/property/ResponseFileName"
                type="xs:string"
                event="http://www.extra-standard.de/event/RequestData">
                <xmsg:EQ>R46934503</xmsg:EQ>
            </xmsg:Argument>

            <xmsg:Argument
                property="http://www.extra-standard.de/property/ResponseCreationTimeStamp"
                type="xs:dateTime">
                <xmsg:GT>2009-03-16T09:00:00</xmsg:GT>
                <xmsg:LE>2009-03-17T09:00:00</xmsg:LE>
            </xmsg:Argument>

        </xmsg:Query>

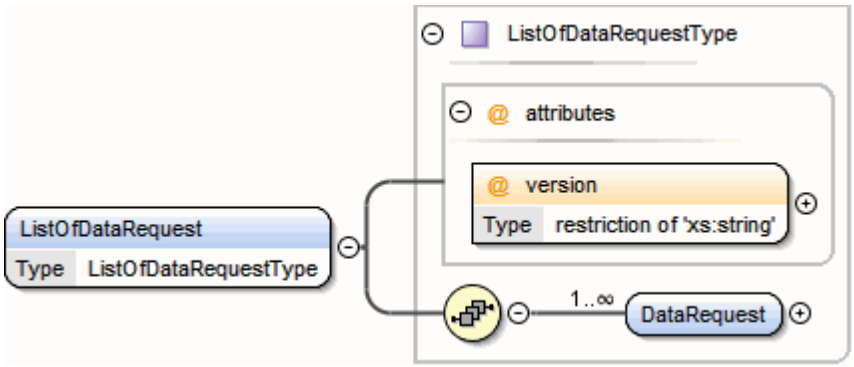
        <xmsg:Control>
            <xmsg:MaximumPackages>10</xmsg:MaximumPackages>
            <xmsg:MaximumMessages>100</xmsg:MaximumMessages>
            <xmsg:MaximumSize>10M</xmsg:MaximumSize>
        </xmsg:Control>

    </xmsg:DataRequest>


</xmsg:ListOfDataRequest>
```

10.2 Element ListOfDataRequest

	`xmsg:ListOfDataRequest`
Inhalt	Eine Menge von Nachrichten des Typs *DataRequest*.
Datentyp	`##elements`
Benutzt von	-



Attribute

	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung dieser eXTra Standardnachricht, derzeit 1.1	
Datentyp	`xsd:string`	

Beschreibung

`xmsg:ListOfDataRequest` ist das Wurzelelement einer eXTra-Nachricht vom Typ DataRequest. Das Attribut `version` enthält die Versionsbezeichnung.

Eine Nachricht dieses Typs dient zur Übertragung einer Menge von Nachrichten des Typs *DataRequest*

11 Nachricht ListOfStatusResponse [normativ]

Eine Nachricht dieses Typs dient zur Übertragung einer Menge von Nachrichten des Typs *StatusResponse*.

Der Name des Wurzelelementes ist `ListOfStatusResponse`.

Zu Namensraum und Präfix siehe Abschnitt *Namensräume und Präfixe*.

11.1 Beispieldokument

```
<?xml version="1.0" encoding="UTF-8"?>
<xmsg:ListOfStatusResponse
    xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    version="1.0"
    >

    <xmsg:StatusResponse>

        <xmsg:Property name="http://www.extra-standard.de/property/RequestID">
            <xmsg:Value>12345</xmsg:Value>
        </xmsg:Property>

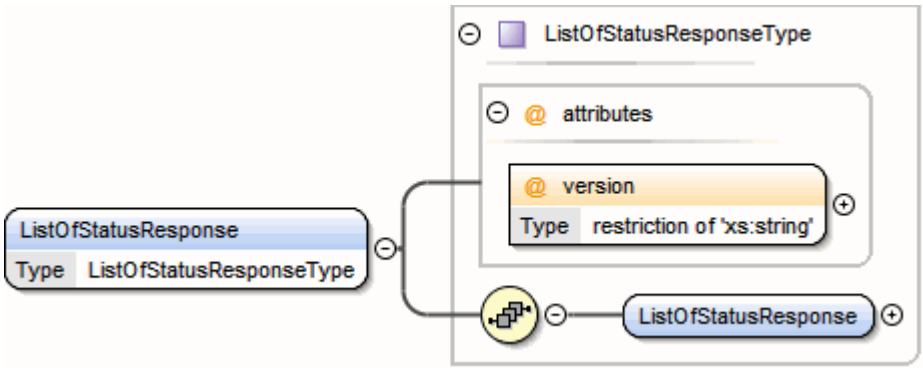
        <xmsg:Trace>
            <xmsg:Checkpoint>
                <xmsg:Layer>http://www.extra-standard.de/layer/Transport</xmsg:Layer>
                <xmsg:Status>http://www.extra-standard.de/status/COMPLETED</xmsg:Status>
                <xmsg:Timestamp>2011-05-08T22:00:10</xmsg:Timestamp>
            </xmsg:Checkpoint>
            <xmsg:Checkpoint>
                <xmsg:Layer>http://www.extra-standard.de/layer/Application</xmsg:Layer>
                <xmsg:Status>http://www.extra-standard.de/status/PROCESSING</xmsg:Status>
                <xmsg:Timestamp>2011-05-08T22:00:10</xmsg:Timestamp>
                <xmsg:Report highestWeight="http://www.extra-standard.de/weight/OK">
                    <xmsg:Flag weight="http://www.extra-standard.de/weight/OK">
                        <xmsg:Code>20</xmsg:Code>
                        <xmsg:Text>Et it noch allet jot jejange!</xmsg:Text>
                    </xmsg:Flag>
                </xmsg:Report>
            </xmsg:Checkpoint>
        </xmsg:Trace>

    </xmsg:StatusResponse>

</xmsg:ListOfStatusResponse>
```

11.2 Element ListOfStatusResponse

	`xmsg:ListOfDataRequest`
Inhalt	Eine Menge von Nachrichten des Typs *StatusResponse*
Datentyp	`##elements`
Benutzt von	-



Attribute

	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung dieser eXTra-Standardnachricht, derzeit 1.0	
Datentyp	`xsd:string`	

Beschreibung

`xmsg:ListOfStatusResponse` ist das Wurzelelement einer eXTra-Nachricht vom Typ *ListOfStatusResponse*. Das Attribut `version` enthält die Versionsbezeichnung.

Eine Nachricht dieses Typs dient zur Übertragung einer Menge von Nachrichten des Typs *StatusResponse*

12 Nachricht RepeatResponse [normativ]

Mit einer Nachricht des Typs *RepeatResponse* wird das erneute Senden einer Response angefordert.

Die Nachricht wird nicht durch eine XML Struktur repräsentiert, sondern entsteht durch Senden eines eXTra-Request Headers, dessen Eigenschaft `DataType` auf den Wert

```
http://www.extra-standard.de/datatype/RepeatResponse
```

gesetzt ist.

13 Nachricht RepeatResponseRequest

Mit einer Nachricht des Typs *RepeatResponseRequest* wird das erneute Senden einer Response angefordert. Im Gegensatz zu RepeatResponse [12] wird die Anfrage mit den Standardsprachmitteln formuliert, wie sie auch in anderen Request-Nachrichten Verwendung finden.

Der Name des Wurzelelementes ist `RepeatResponseRequest`.

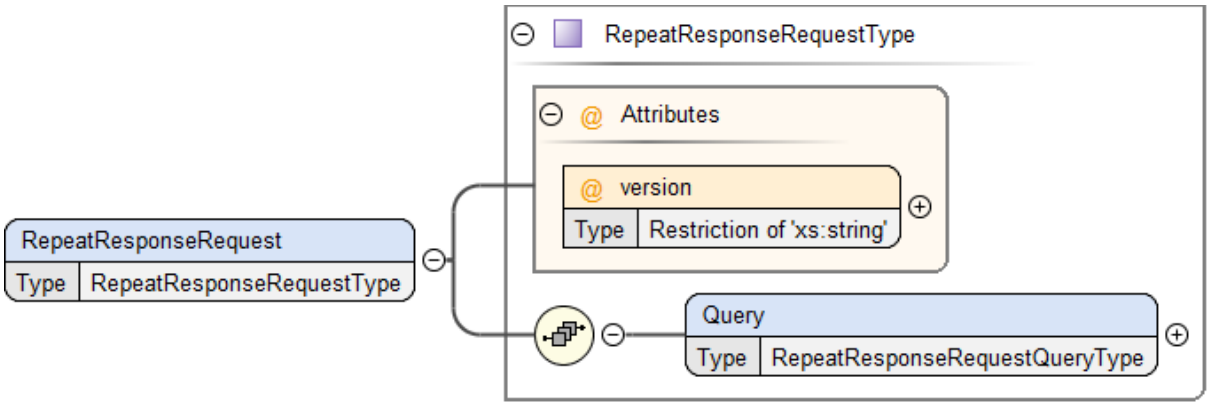
Zu Namensraum und Präfix siehe Abschnitt *Namensräume und Präfixe*.

13.1 Beispieldokument

```
<?xml version="1.0" encoding="UTF-8"?>
<xmsg:RepeatResponseRequest
    xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.0"
    >
    <xmsg:Query>
        <xmsg:Argument
            property=http://www.extra-standard.de/property/DataType
             type="xs:NMTOKEN">
            <xmsg:EQ>AAAAAA</xmsg:EQ>
        </xmsg:Argument>
        <xmsg:Argument
           property="http://www.extra-standard.de/property/RequestID">
            <xmsg:EQ>12345</xmsg:EQ>
        </xmsg:Argument>
        <xmsg:Argument
           property="http://www.extra-standard.de/property/RequestCreationTimeStamp">
            <xmsg:EQ>2011-05-08T22:00:00</xmsg:EQ>
        </xmsg:Argument>
        <xmsg:Argument
           property="http://www.w3.org/haegarthehorrible">
            <xmsg:EQ>aaaa</xmsg:EQ>
        </xmsg:Argument>
    </xmsg:Query></xmsg:RepeatResponseRequest>
```

13.2 Element RepeatResponseRequest

	`xmsg:RepeatResponseRequest`
Inhalt	Eine Nachricht des Typs *RepeatResponseRequest*
Datentyp	`##elements`
Benutzt von	-



Attribute

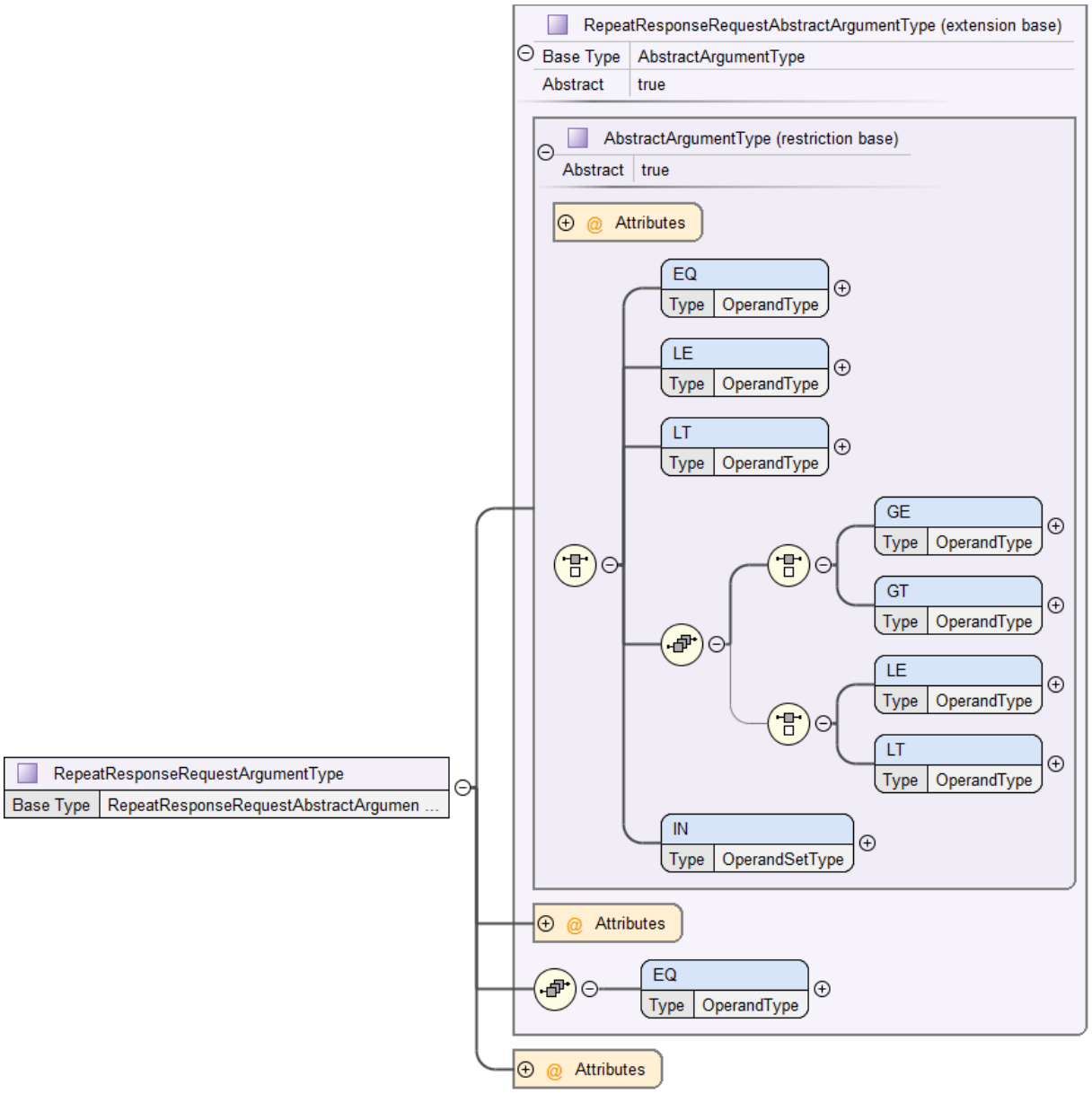
	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung dieser eXTra-Standardnachricht, derzeit 1.0	
Datentyp	`xsd:string`	

Beschreibung

`xmsg: RepeatResponseRequest` ist das Wurzelelement einer eXTra-Nachricht vom Typ *RepeatResponseRequest*. Das Attribut `version` enthält die Versionsbezeichnung.

13.3 Element Argument

	`xmsg:Argument`
Inhalt	Beschreibt ein Kriterium einer Abfrage
Datentyp	`##elements`
Benutzt von	`xmsg:Query`



Attribute

	@property	*erforderlich*
Inhalt	Bezeichnet die abgefragte Eigenschaft	
Datentyp	`xsd:anyURI`	
Werteraum	`##codelist:StatusRequestPropertyNames`	

	@type	*optional*
Inhalt	Der XSD Datentyp der abgefragten Eigenschaft	
Datentyp	`xsd:string`	
Werteraum	`##codelist:ExtraLayerNames`	
Default	`xsd:string`	

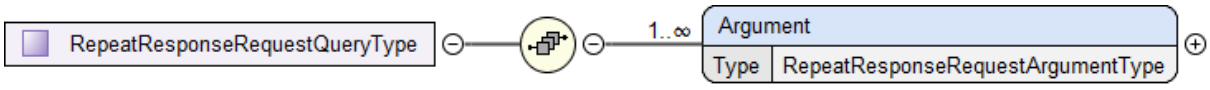
Beschreibung

Ein Kriterium besteht aus einer abgefragten Eigenschaft (Attribut `property`), die per URI referenziert wird, und einem Werteraum, mit dem die Werte der abgefragten Objekte verglichen werden. Der Werteraum wird als Einzelwert oder als Wertebereich mit inklusiver Unter- und Obergrenze oder als Werteliste angegeben. Die wertetragenden Elemente des Werteraums bestimmen zugleich den Operator für den Vergleich mit Einzelwerten bzw. der Unter- oder Obergrenze.

Das Attribut `type` spezifiziert den Datentyp des Arguments.

13.4 Element Query

	`xmsg:Query`
Inhalt	Beschreibt den Abfrageteil der Anforderung.
Datentyp	`##elements`
Benutzt von	`xmsg:StatusRequest`



Beschreibung

Beschreibt eine Abfrage als ein Set von Kriterien (Element `Argument`). Jedes Kriterium ist mit einem Typ behaftet, spezifiziert einen oder mehrere Vergleichswerte und kann auf ein spezifisches Ereignis in einem eXTra-basierten Prozess bezogen werden.

14 Nachricht StatusRequest [normativ]

Mit einer Nachricht des Typs *StatusRequest* kann der Verarbeitungszustand ausgewählter eXTra-Requests abgefragt werden.

Der Name des Wurzelelementes ist `StatusRequest`.

Zu Namensraum und Präfix siehe Abschnitt *Namensräume und Präfixe*.

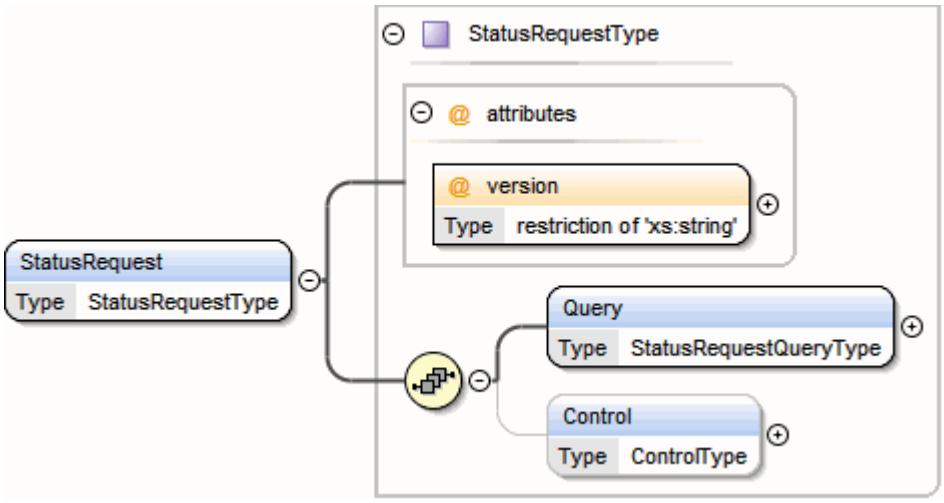
14.1 Beispieldokument

```
<?xml version="1.1" encoding="UTF-8"?>
<xmsg:StatusRequest
    xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    version="1.0"
    >
    <xmsg:Query>
        <xmsg:Argument property=http://www.extra-standard.de/property/DataType
        order="desc">>
            <xmsg:EQ>AAAAAA</xmsg:EQ>
        </xmsg:Argument>
        <xmsg:Argument property="http://www.extra-standard.de/property/RequestID">
            <xmsg:EQ>12345</xmsg:EQ>
        </xmsg:Argument>
        <xmsg:Argument property=
                "http://www.extra-standard.de/property/RequestCreationTimeStamp">
            <xmsg:EQ>2011-05-08T22:00:00</xmsg:EQ>
        </xmsg:Argument>
        <xmsg:Argument property="http://www.w3.org">
            <xmsg:EQ>aaaa</xmsg:EQ>
        </xmsg:Argument>
    </xmsg:Query>

</xmsg:StatusRequest>
```

14.2 Element StatusRequest

	`xmsg:StatusRequest`
Inhalt	Eine Nachricht des Typs *StatusRequest*
Datentyp	`##elements`
Benutzt von	-



Attribute

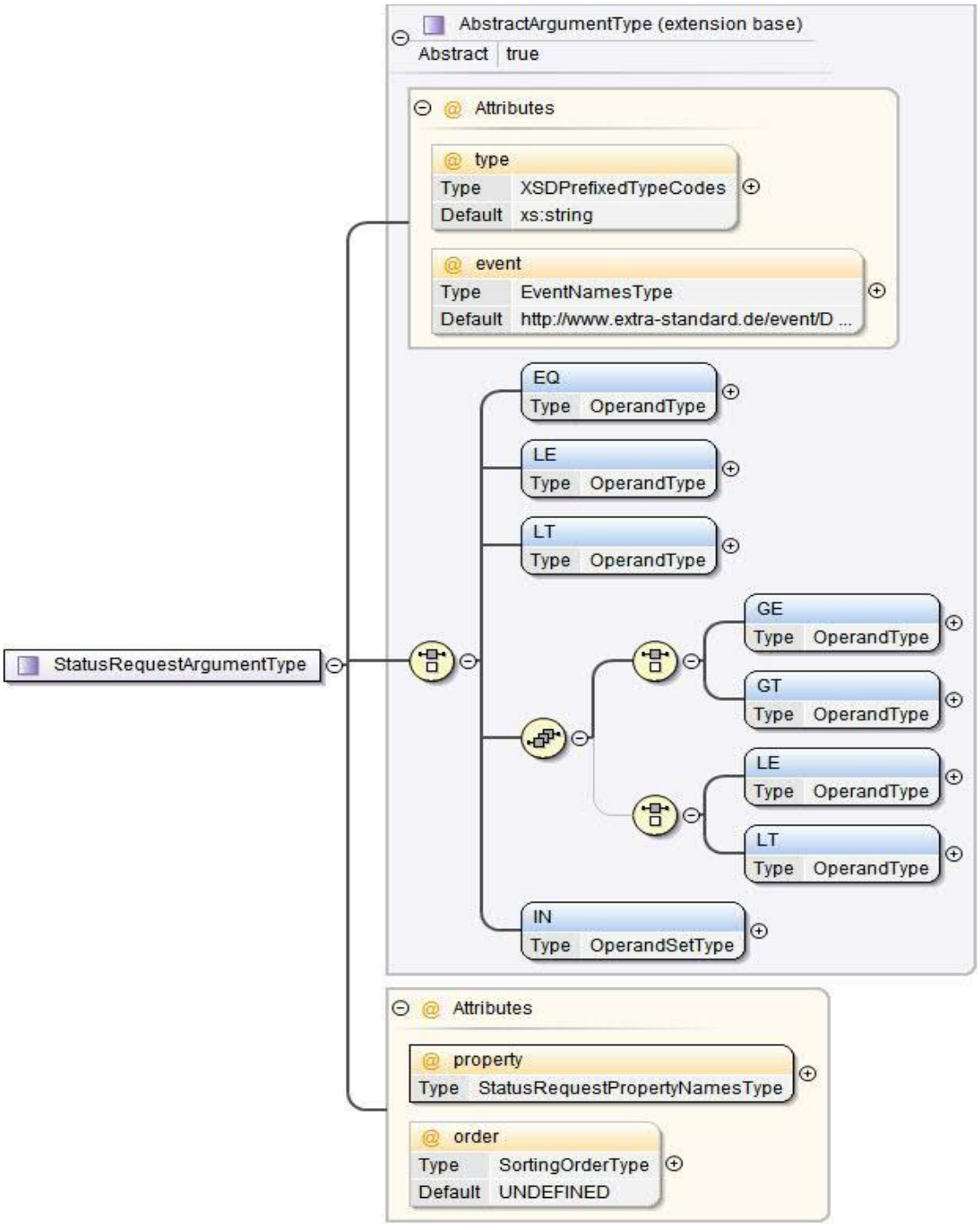
	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung dieser eXTra-Standardnachricht, derzeit 1.1	
Datentyp	`xsd:string`	

Beschreibung

`xmsg:StatusRequest` ist das Wurzelelement einer eXTra-Nachricht vom Typ *StatusRequest*. Das Attribut `version` enthält die Versionsbezeichnung.

14.3 Element Argument

	xmsg:Argument
Inhalt	Beschreibt ein Kriterium einer Abfrage
Datentyp	`##elements`
Benutzt von	`xmsg:Query`



Attribute

	@event	*optional*
Inhalt	Bezieht die abgefragte Eigenschaft auf ein Ereignis im Kommunikationsablauf	
Datentyp	`xsd:anyURI`	
Werteraum	`##codelist:EventNames`	
Default	`http:www.extra-standard.de/event/Default`	

	@order	*optional*
Inhalt	Bezeichnet die gewünschte Sortierung der Elemente der Antwort auf die Anfrage	
Werteraum	`##codelist:SortingOrder`	
Datentyp	`xsd:string`	

	@property	*erforderlich*
Inhalt	Bezeichnet die abgefragte Eigenschaft	
Datentyp	`xsd:anyURI`	
Werteraum	`##codelist:StatusRequestPropertyNames`	

	@type	*optional*
Inhalt	Der XSD Datentyp der abgefragten Eigenschaft	
Datentyp	`xsd:string`	
Werteraum	`##codelist:ExtraLayerNames`	
Default	`xsd:string`	

Profilierung

Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
@event						+

Beschreibung

Ein Kriterium besteht aus einer abgefragten Eigenschaft (Attribut `property`), die per URI referenziert wird, und einem Werteraum, mit dem die Werte der abgefragten Objekte verglichen werden. Der Werteraum wird als Einzelwert oder als Wertebereich mit inklusiver Unter- und Obergrenze oder als Werteliste angegeben. Die wertetragenden Elemente des Werteraums bestimmen zugleich den Operator für den Vergleich mit Einzelwerten bzw. der Unter- oder Obergrenze.

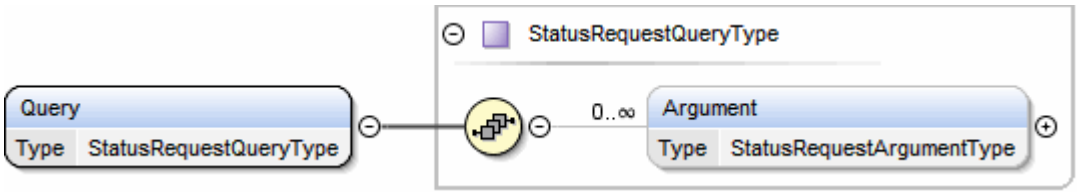
Das Attribut `type` spezifiziert den Datentyp des Arguments.

Mit dem Attribut `event` wird das Ereignis im Kommunikationsablauf spezifiziert, für das die Eigenschaft abgefragt werden soll. Die vordefinierte Werteliste kann durch nutzerspezifische URIs ersetzt bzw. ergänzt werden.

Die Reihenfolge der auszuliefernden Elemente der Antwort auf die Anfrage kann mit dem Attribut `order` festgelegt werden.

14.4 Element Query

	`xmsg:Query`
Inhalt	Beschreibt den Abfrageteil der Anforderung.
Datentyp	`##elements`
Benutzt von	`xmsg:StatusRequest`



Beschreibung

Beschreibt eine Abfrage als ein Set von Kriterien (Element `Argument`). Jedes Kriterium ist mit einem Typ behaftet, spezifiziert einen oder mehrere Vergleichswerte und kann auf ein spezifisches Ereignis in einem eXTra-basierten Prozess bezogen werden.

15 Nachricht StatusResponse [normativ]

Mit einer Nachricht des Typs *StatusResponse* wird das Ergebnis einer Statusabfrage über den Verarbeitungszustand einer fachlichen Nachricht übertragen. Mengen von Nachrichten dieses Typs können mit *ListOfStatusResponse* gebündelt werden. Statusabfragen erfolgen typischerweise per Nachrichten des Type *StatusRequest*.

Der Name des Wurzelelementes ist `StatusResponse`.

Zu Namensraum und Präfix siehe Abschnitt *Namensräume und Präfixe*.

15.1 Beispieldokument

```
<?xml                              version="1.0"                              encoding="UTF-8"?>
<xmsg:StatusResponse
    xmlns:xmsg="http://www.extra-standard.de/namespace/message/1"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    version="1.0"
    >

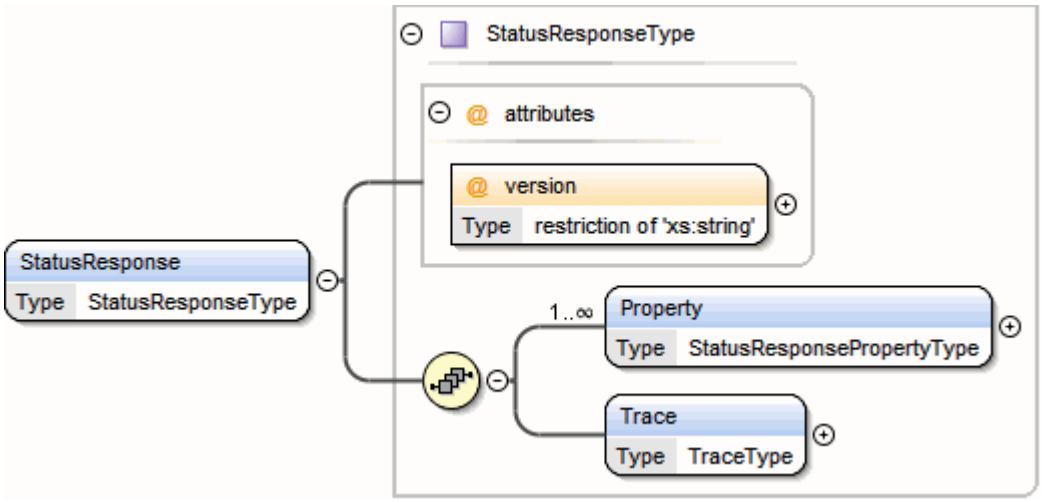
    <xmsg:Property                      name="http://www.extra-standard.de/property/RequestID">
        <xmsg:Value>12345</xmsg:Value>
    </xmsg:Property>

    <xmsg:Trace>
        <xmsg:Checkpoint>
            <xmsg:Layer>http://www.extra-standard.de/layer/Transport</xmsg:Layer>
            <xmsg:Status>http://www.extra-standard.de/status/COMPLETED</xmsg:Status>
            <xmsg:Timestamp>2011-05-08T22:00:10</xmsg:Timestamp>
        </xmsg:Checkpoint>
        <xmsg:Checkpoint>
            <xmsg:Layer>http://www.extra-standard.de/layer/Application</xmsg:Layer>
            <xmsg:Status>http://www.extra-standard.de/status/PROCESSING</xmsg:Status>
            <xmsg:Timestamp>2011-05-08T22:00:10</xmsg:Timestamp>
            <xmsg:Report               highestWeight="http://www.extra-standard.de/weight/OK">
                <xmsg:Flag                      weight="http://www.extra-standard.de/weight/OK">
                    <xmsg:Code>20</xmsg:Code>
                    <xmsg:Text>Et      it      noch      allet      jot      jejange!</xmsg:Text>
                </xmsg:Flag>
            </xmsg:Report>
        </xmsg:Checkpoint>
    </xmsg:Trace>

</xmsg:StatusResponse>
```

15.2 Element StatusResponse

	`xmsg:StatusResponse`
Inhalt	Eine Nachricht des Typs *StatusResponse*
Datentyp	`##elements`
Benutzt von	-



Attribute

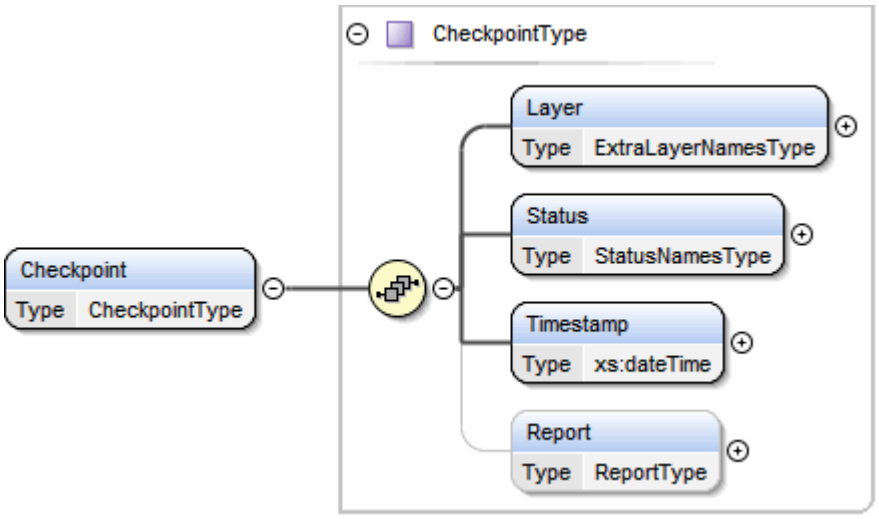
	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung dieser eXTra-Standardnachricht, derzeit 1.0	
Datentyp	`xsd:string`	

Beschreibung

`xmsg:StatusResponse` ist das Wurzelelement einer eXTra-Nachricht vom Typ *Status-Response*. Das Attribut `version` enthält die Versionsbezeichnung.

15.3 Element Checkpoint

	`xmsg:Checkpoint`
Inhalt	Checkpoint für einen definierten Punkt der Prozesskette
Datentyp	`##elements`
Benutzt von	`xmsg:Trace`

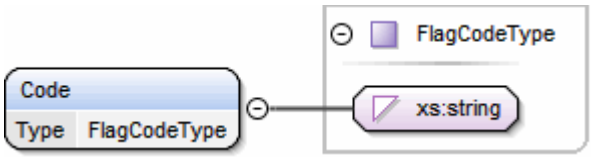


Beschreibung

Ein Checkpoint beschreibt den Verarbeitungszustand eines eXTra-Request an einem definierten Punkt der Prozesskette.

15.4 Element Code

	xmsg:Code
Inhalt	Ein Schlüssel, der eine kodierte Information darstellt, z.B. ein Fehlercode
Datentyp	`xsd:string`
Benutzt von	`xmsg:Flag`

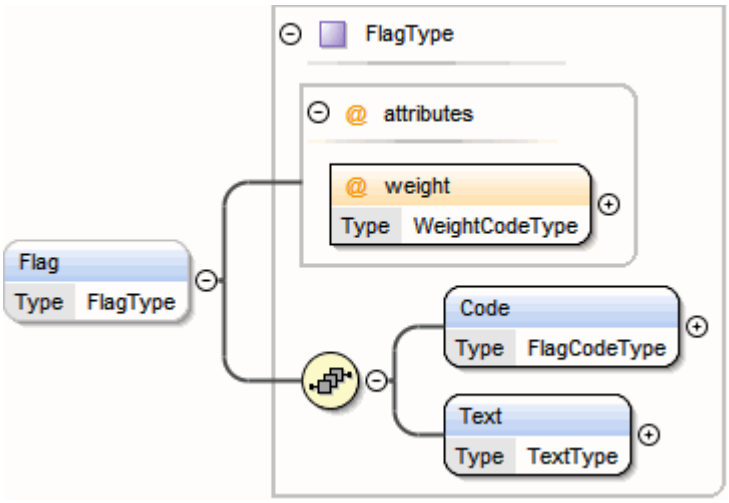


Beschreibung

`xmsg:Code` dient zur Angabe eines Fehlerschlüssels oder einer anderen kodierten Information in Kombination mit einer durch `xmsg:Text` dargestellten textuellen Ausprägung.

15.5 Element Flag

	xmsg:Flag
Inhalt	Flag mit Gewicht, Code und Text
Datentyp	`##elements`
Benutzt von	`xmsg:Report`



Attribute

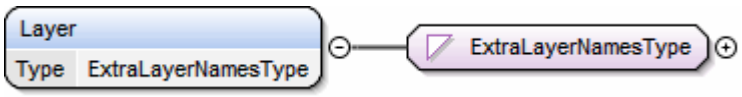
	@weight	*erforderlich*
Inhalt	Gewicht des Flags	
Datentyp	`xsd:anyURI`	
Werteraum	`##codelist:XSDWeightCodes`	

Beschreibung

Flags repräsentieren Verarbeitungsergebnisse durch einen formalen Code, einen erläuternden Text und ein Fehlergewicht.

15.6 Element Layer

	xmsg:Layer
Inhalt	Spezifiziert eine Ebene des eXTra-Kommunikationsmodells.
Datentyp	`xsd:anyURI`
Werteraum	`##codelist:ExtraLayerNames`
Benutzt von	`xmsg:Checkpoint`



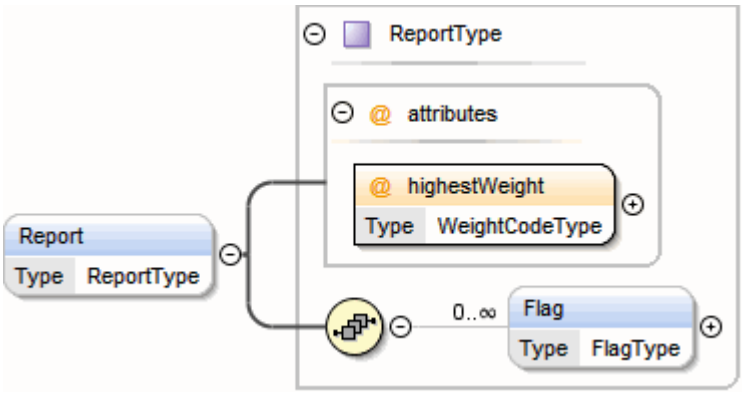
Beschreibung

`xmsg:Layer` spezifiziert in einem Checkpoint die Ebene des eXTra-Kommunikationsmodells.

Die Werte sind abschließend in der Codelist *ExtraLayerNames* definiert.

15.7 Element Report

	xmsg:Report
Inhalt	Enthält alle Flags für den gegebenen Checkpoint
Datentyp	`##elements`
Benutzt von	`xmsg:Checkpoint`



Attribute

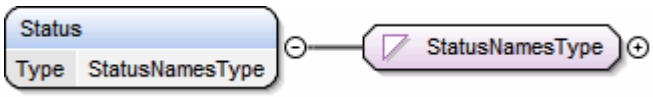
	@highestWeight	*erforderlich*
Inhalt	Das höchste vergebene Fehlergewicht aller Flags des Reports.	
Datentyp	`xsd:anyURI`	
Werteraum	`##codelist:XSDWeightCodes`	

Beschreibung

Ein Report besteht aus einer Menge von Flags, die Informationen, Warnungen oder Fehler beschreiben.

15.8 Element Status

	xmsg:Status
Inhalt	Spezifiziert den Status der Verarbeitung an einem Checkpoint.
Datentyp	`xsd:anyURI`
Werteraum	`##codelist:StatusNames`
Benutzt von	`xmsg:Checkpoint`

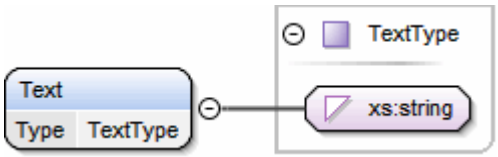


Beschreibung

`xmsg:Status` spezifiziert in einem Checkpoint den Verarbeitungsstatus. Die Werte sind in der Codelist *StatusNames* definiert.

15.9 Element Text

	xmsg:Text
Inhalt	Ein Text
Datentyp	xsd:string
Benutzt von	xmsg:Flag

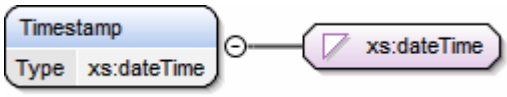


Beschreibung

xmsg:Text enthält in Verbindung mit xmsg:Code die textuelle Ausprägung einer kodierten Information wie z.B. eines Fehlerschlüssels..

15.10 Element Timestamp

	`xmsg:Timestamp`
Inhalt	Ein Zeitstempel im ISO Format
Datentyp	`xsd:dateTime`
Benutzt von	`xmsg:Checkpoint`

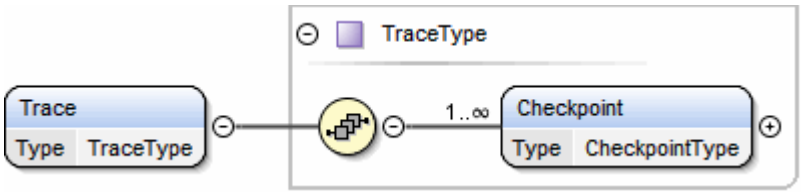


Beschreibung

`xmsg:Timestamp` beschreibt den Zeitpunkt, an dem ein eXTra-Request an dem durch `xmsg:Checkpoint` definierten Punkt der Prozesskette angekommen ist.

15.11 Element Trace

	`xmsg:Trace`
Inhalt	Eine Folge von Checkpoints
Datentyp	`##elements`
Benutzt von	`xmsg:StatusResponse`



Beschreibung

`xmsg:Trace` enthält eine Folge von Checkpoints. Jeder Checkpoint beschreibt den Verarbeitungsstatus eines Teilprozesses in einer eXTra-basierten Prozesskette.

16 Ressourcen

16.1 XSD Schema Ressourcen

Das eXTra-XSD-Schema für Standardnachrichten besteht aus den folgenden Schemadateien:

`eXTra-messages-1.xsd`	eXTra-Standardnachrichten

17 Referenzen

Kurzname	Quelle
BEST	*eXTra Best Practices* zu finden unter www.extra-standard.de
DSIG	*eXTra Design Guidelines*, zu finden unter www.extra-standard.de
EINF	*Einführung in den eXTra-Standard*, zu finden unter www.extra-standard.de
EMSG	*eXTra-Standardnachrichten, Schnittstellenbeschreibung*, zu finden unter www.extra-standard.de
EXSEC	*Sicherheit und Verfügbarkeit in einem eXTra spezifischen Datenübermittlungsverbund*, zu finden unter www.extra-standard.de
EXWS	*eXTra und Webservices*, zu finden unter www.extra-standard.de
IFACE	*eXTra-Transport Schnittstellenbeschreibung*, zu finden unter www.extra-standard.de
KOMP	*eXTra-Kompendium*, zu finden unter www.extra-standard.de
RFC2119	*Request for Comments: 2119*, S. Bradner, Harvard University, March 1997, http://www.ietf.org/rfc/rfc2119.txt
PROF	*eXTra-Profilierung,* zu finden unter www.extra-standard.de
ÜMSG	*eXTra Standardnachrichten, Überblick*, zu finden unter www.extra-standard.de
VERS	*eXTra Versionierung*, zu finden unter www.extra-standard.de
XENC	*XML Encryption*, http://www.w3.org/TR/xmlenc-core/
XML	*XML Recommendation 1.0, 3rd Edition*, http://www.w3.org/XML
XSD	*XML Schema Definition*, http://www.w3.org/TR/xmlschema-0/
XSIG	*XML Signature*, http://www.w3.org/TR/xmldsig-core/
XSL	*XML Stylesheet Language*, http://www.w3.org/TR/1999/REC-xslt-19991116, http://www.w3.org/TR/xslt20/