

einheitliches XML-basiertes Transportverfahren

eXTra-Transport Version 1.4 Schnittstellenbeschreibung

Ausgabestand 1.4.0

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 von Version 1.3.0 auf Version 1.4.0

Autor[en]	Datum	Beschreibung
[gelöscht]	05.10.2013	Unterstützung großer Datenmengen mittels MTOM Neue Schemadatei Components mit erweitertem Element Base64CharSequence Erweiterung bewirkt keinen Änderungsbedarf bei bestehenden XML-Dokumenten der V1.3

Änderungsprotokoll von Version 1.2.1 auf Version 1.3.0

Autor[en]	Datum	Beschreibung
[gelöscht]	21.10.2011	Neue Codeliste *DataTypeCodes* Neues alternatives Root Element `Transport`

Änderungsprotokoll von Version 1.1 auf Version 1.2.1

Autor[en]	Datum	Beschreibung
[gelöscht]	29.07.2010	Übernahme und Anpassungen für die Version 1.2 des eXTra Basis Standards. Neue Nachricht `ExtraError` Neues PlugIn `Contact` PlugIn `DataTransforms`: Der Signatur kann jetzt das Zertifikat beigelegt werden.. Neues Kapitel „Versionierung“

Änderungsprotokoll für Ausgabestand 1.1.0 bis 1.1.2

Autor[en]	Datum	Beschreibung
[gelöscht]	29.07.2009	Ersterstellung
	08.03.2010	Freigabe

Inhaltsverzeichnis

1 Einleitung 7
1.1 Ausgabestand 7
1.2 Verwendung 7
1.3 Der eXTra-Standard 7
1.4 Begriffe 9
2 Layout und Notationen 11
2.1 Referenzen 11
2.2 Formatierungen 11
2.3 Beschreibung der Regeln 11
2.4 Beschreibung der Elemente und Attribute 12
2.5 Graphische Symbole 13
3 Aufbau von eXTra Dokumenten [normativ] 16
3.1 Namensräume 16
3.2 Grundstrukturen 17
3.3 Versionierung 19
4 eXTra Codelisten [normativ] 21
4.1 Codeliste CompressionAlgorithmCode 22
4.2 Codeliste DataContainerCode 22
4.3 Codeliste DataTypeCode 22
4.4 Codeliste EncryptionAlgorithmCode 23
4.5 Codeliste OperationCode 23
4.6 Codeliste ScenarioCode 24
4.7 Codeliste SignatureAlgorithmCode 24
4.8 Codeliste TestIndicatorCode 24
4.9 Codeliste UsageCode 25
4.10 Codeliste WeightCode 25
4.11 Codeliste XSDTypeCodes 25
5 Nachrichtentyp Request [normativ] 26
5.1 Element Transport 27
5.2 Element XMLTransport 29
5.3 Element Message 30
5.4 Element MessageBody 31
5.5 Element MessageHeader 32
5.6 Element MessagePlugIns 33
5.7 Element Package 34
5.8 Element PackageBody 35
5.9 Element PackageHeader 36
5.10 Element PackagePlugIns 37
5.11 Element TransportBody 38
5.12 Element TransportHeader 40
5.13 Element TransportPlugIns 41
6 Nachrichtentyp Response [normativ] 42
6.1 Element Transport 43
6.2 Element XMLTransport 45
6.3 Element Message 46
6.4 Element MessageBody 48
6.5 Element MessageHeader 49
6.6 Element MessagePlugIns 50
6.7 Element Package 51
6.8 Element PackageBody 52
6.9 Element PackageHeader 53

6.10 Element PackagePlugIns 54
6.11 Element TransportBody 55
6.12 Element TransportHeader 57
6.13 Element TransportPlugIns 58
7 eXTra-Error [normativ] 59
7.1 Element ExtraError 60
7.2 Element Reason 61
8 Plug-Ins [normativ] 62
8.1 Element Value 63
8.2 Element X509Certificate 64
8.3 Plug-In Certificates 65
8.4 Element X509EncCertificate 66
8.5 Plug-In Contacts 67
8.6 Element SenderContact 68
8.7 Plug-In DataSource 69
8.8 Element DataContainer 70
8.9 Element DataSet 72
8.10 Plug-In DataTransforms 73
8.11 Element Compression 75
8.12 Element Compression/Algorithm 76
8.13 Element Encryption 77
8.14 Element Encryption/Algorithm 78
8.15 Element InputData 79
8.16 Element OutputData 80
8.17 Element Signature 81
8.18 Element Signature/Algorithm 82
8.19 Element Specification 83
9 Logging [normativ] 84
9.1 Element Logging 85
9.2 Element Event 87
9.3 Element Exception 88
9.4 Element LogID 89
9.5 Element LogSequence 90
9.6 Element LogStream 91
9.7 Element LogTrace 92
9.8 Element Message 93
9.9 Element Object 94
9.10 Element Operation 96
9.11 Element Parameter 98
9.12 Element PreviousLogID 100
9.13 Element StackTrace 101
9.14 Element State 102
10 Allgemeine Komponenten [normativ] 103
10.1 Element AnyXML 104
10.2 Element Application 105
10.3 Element Base64CharSequence 106
10.4 Element CharSequence 107
10.5 Element Code 108
10.6 Element ComponentID 109
10.7 Element Data 110
10.8 Element DataType 111
10.9 Element ElementSequence 112
10.10 Element Flag 113
10.11 Element Manufacturer 115

10.12 Element Name 116
10.13 Element Originator 117
10.14 Element Procedure 118
10.15 Element ProcessID 119
10.16 Element Product 120
10.17 Element Property 120
10.18 Element Receiver 121
10.19 Element ReceiverID 123
10.20 Element RegistrationID 124
10.21 Element Report 125
10.22 Element RequestDetails 127
10.23 Element RequestID 128
10.24 Element ResponseDetails 129
10.25 Element ResponseID 130
10.26 Element Scenario 131
10.27 Element Sender 132
10.28 Element SenderID 133
10.29 Element Signatures 134
10.30 Element Stack 135
10.31 Element TestIndicator 136
10.32 Element Text 138
10.33 Element TimeStamp 139
10.34 Element TransformedData 140
10.35 Element XMLFragment 141
10.36 Element XPath 143
11 Ressourcen 144
11.1 XSD Schema Ressourcen 144
12 Literatur 145

1 Einleitung

1.1 Ausgabestand

Dieses Dokument mit dem Ausgabestand 1.4 enthält die normative technische Spezifikation der Version 1.4 der Transport-Schnittstelle des eXTra-Standards.

In der Versionsangabe dieses Dokuments V *n.m.r* ist in den ersten beiden Ziffern n.m die jeweilige Version der zugrundeliegenden Transport-Schnittstelle des eXTra-Basisstandards angeführt, während die letzte Ziffer *r* den jeweiligen Ausgabestand wiedergibt.

1.2 Verwendung

Die Schnittstellenspezifikation kann auf zweierlei Weise verwendet werden:

- In der PDF-Version als normative Schnittstellenbeschreibung.
- In der Word-Version als Vorlage für die Erstellung eigener, profilierter Schnittstellenbeschreibungen.

1.3 Der eXTra-Standard

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 zu eXTra sind im Internet unter der Adresse http://www.extra-standard.de abrufbar.

Diese Spezifikation beschreibt Strukturen und Felder von eXTra-Dokumenten. 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 Implementierung der gesamten eXTra-Kommunikationsschnittstelle, die aus zwei Teilen bestehen kann, obligatorisch aus der eXTra-Transport-Schnittstelle und optional ergänzt um die Schnittstelle der *eXTra-Standardnachrichten* [XMSG]. Diese Dokumente richten sich vor allem an Anwendungsarchitekten und Entwickler und sind Ausgangspunkt für eine evtl. sich anschließende Profilierung der eXTra-Kommunikationsschnittstelle, die in [PROF] erläutert ist.

eXTra umfasst also neben der in diesem Dokument beschriebenen unprofilierten Transportschicht eine Reihe sogenannter Standardnachrichten. Standardnachrichten sind als Nutzdaten zu betrachten und wie diese zu behandeln. Aus diesem Grund befinden sich alle Standardnachrichten in einem gemeinsamen, separaten Namensraum und haben keinerlei Abhängigkeit von den Schemata der Transportschicht. Wie die Transportschicht können die Standardnachrichten profiliert werden.

Standardnachrichten decken einer Reihe typischer Transaktionen in Datenaustauschszenarien ab, wie z.B. das Anfordern von Daten und Empfangsbestätigungen. Sie stellen damit eine weitere Vereinfachung bei der Einführung und Verwendung von eXTra dar.

Die eXTra-Kommunikationsschnittstelle basiert auf der *Extensible Markup Language – XML* – in der Version 1.0, wie beschrieben durch [XML]. eXTra-Nachrichten 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-Nachrichtentypen 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 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 Kon-

strukte 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-Nachrichten. 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. Elemente aus von Dritten definierten Namensräumen wie XML Encryption [XENC] und XML Signature [XSIG] sind in der entsprechenden öffentlichen Spezifikation dieser Namensräume beschrieben.

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

1.4 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]** gekennzeichneten Kapiteln 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 ande-

rer 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 können Fontgewicht und Fontgröße von diesen Angaben abweichen.

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* den 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*.

Folgende Profilierungsoptionen werden in eXTra unterstützt (in [] der Name der Spalte):

[*REQ*]	Profilierung einer Komponente als erforderlich (*required*)
[*OPT*]	Profilierung einer Komponente als optional (*optional*)
[*FOR*]	Profilierung einer Komponente als verboten (*forbidden*)
[*MUL*]	Profilierung der Kardinalität einer Komponente (*multiplicity*)
[*IGN*]	Profilierung einer Komponente als zu ignorieren (*ignore*)
[*VAL*]	Profilierung des Wertebereiches (*values*) einer Komponente (nur Elemente mit Dateninhalt und Attribute)

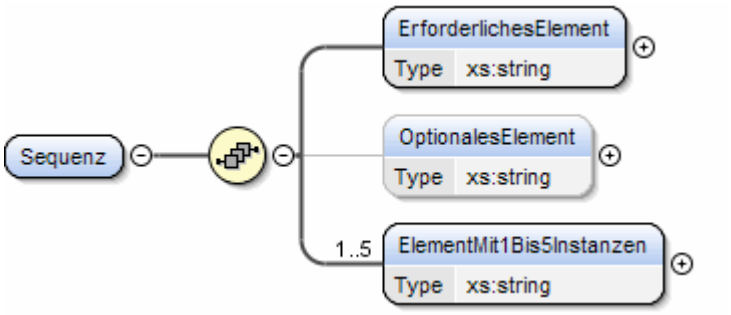
2.5 Graphische Symbole

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

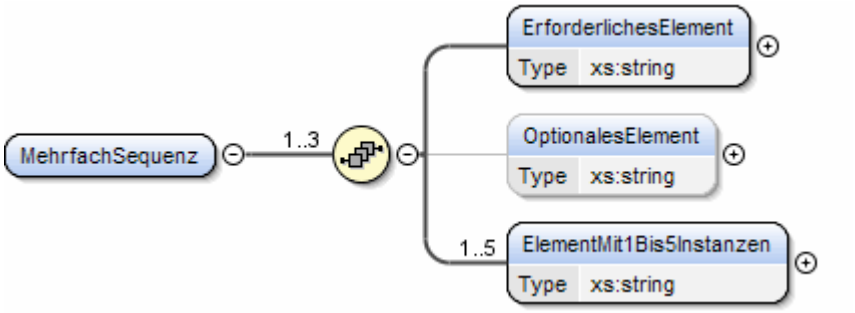
Hinweis: In den meisten Fällen erscheinen Elemente und Attribute in einem im Farbdruck gelb eingefärbten Rechteck. Dieses stellt eine XSD-Typdefinition dar und trägt deren Namen. Typdefinitionen sind schemaspezifisch und daher nicht normativ. Aus diesem Grund findet sich in den folgenden Mustergraphiken 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.

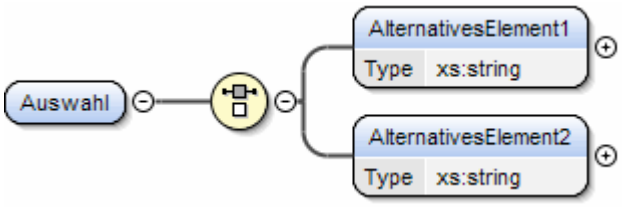
Eine Sequenz von Elementen



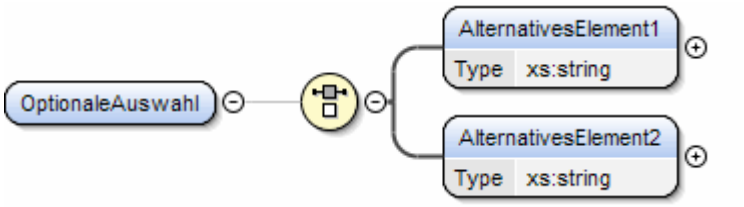
Eine bis zu dreimal wiederholte Sequenz von Elementen



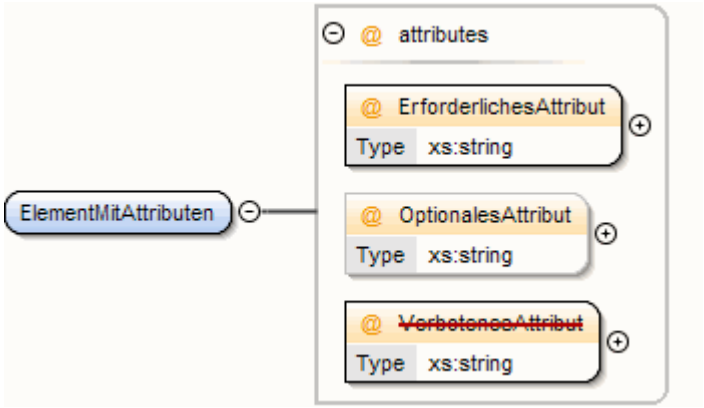
Eine erforderliche Auswahl (Kindelemente sind Pflichtelemente)



Eine optionale Auswahl von (Kindelemente sind Pflichtelemente)



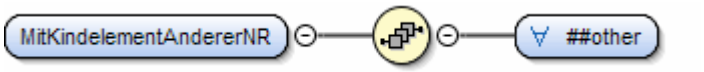
Ein Element mit Attributen



Ein Element mit einem beliebigen Kindelement aus dem eigenen Namensraum



Ein Element mit einem beliebigen Kindelement aus einem anderen Namensraum



---

3 Aufbau von eXTra Dokumenten [normativ]

3.1 Namensräume

eXTra-Transport verwendet die folgenden Namensräume und Präfixe. Die Verwendung einheitlicher Präfixe für einen Namensraum ist eine XML Best Practice, und es wird daher EMPFOHLEN, nur die genannten Präfixe zu verwenden.

Namensraum	Präfix
http://www.extra-standard.de/namespace/request/1	xreq, xtra
http://www.extra-standard.de/namespace/response/1	xres, xtra
http://www.extra-standard.de/namespace/components/1	xcpt
http://www.extra-standard.de/namespace/codelists/1	xcode
http://www.extra-standard.de/namespace/plugin/1	xplg
http://www.extra-standard.de/namespace/logging/1	xlog
http://www.extra-standard.de/namespace/service/1	xsrv
http://www.w3.org/2001/04/xmlenc#	xenc
http://www.w3.org/2000/09/xmldsig#	ds

Nachfolgend werden Namensräume mit ihrem Kurznamen referenziert, der in der URI der Versionsnummer vorangeht, z.B. *request* oder *codelists*.

3.2 Grundstrukturen

eXTra verwendet Namensräume, um Nachrichten funktional zu unterscheiden und strukturell zu ordnen und zu modularisieren. Dabei kommen die folgenden prinzipiellen Regeln zur Anwendung:

Statische Aspekte

Im Basisausbau besteht ein eXTra-Request aus den Namensräumen *request* und *components* und eine eXTra-Response aus den Namensräumen *response* und *components*. Die Nutzung anderer eXTra-Namensräume wie *plugins* und *logging* ist abhängig von der Nutzung der entsprechenden Sprachmittel, welche mittels Profilierung festgelegt wird.

Die Namensräume *request* und *response* definieren mit den Wurzelelementen und den Ebenen und ihren Headern die Grundstruktur eines Request bzw. einer Response. Im Rahmen der Profilierung kann die Anzahl der eXTra-Ebenen eingeschränkt und der Aufbau der einzelnen Ebenen sowie der eXTra-Header bedarfsgerecht festgelegt werden. Dies kann für Request und Response auf unterschiedliche Weise erfolgen. So kann z.B. ein Request aus drei Ebenen und die entsprechende Response nur aus einer Ebene bestehen, etwa, wenn sie nur als Empfangsbestätigung auf Transportebene dient [DSIG].

Der Namensraum *components* stellt einheitliche Definitionen für übergreifend verwendete Strukturen zur Verfügung.

Dynamische Aspekte

Im dynamischen Prozessablauf legt der Sender durch Angabe eines Kommunikationsszenarios fest, ob er überhaupt eine Response, eine Response mit Ergebnisinformationen lediglich des eXTra Empfangssystems oder eine Response mit Ergebnisinformationen sowohl des eXTra Empfangssystems als auch des Fachverfahrens erwartet. Das Kommunikationsszenario wird im Header des Request im Kindelement `scenario` des Elements `RequestDetails` eingetragen. Hierbei bedeuten:

- `request-with-acknowledgement`: Es wird - im Fall eines erfolgreichen Transports - nur eine Empfangsbestätigung des eXTra-Empfangssystems erwartet, die fachliche Verarbeitung der Daten durch das Fachverfahren und die Übermittlung der fachlichen Antwort erfolgen asynchron. Die Ergebnisinformationen des eXTra-

Empfangssystems werden im Header der Response im Kindelement `report` des Elements `ResponseDetails` eingetragen

- `request-with-response`: Die Response soll neben den Ergebnisinformationen des eXTra-Empfangssystems bereits das Ergebnis der fachlichen Verarbeitung durch das Fachverfahren enthalten. Die entsprechenden fachlichen Daten werden im Body der untersten Ebene der Response, die Ergebnisinformationen des eXTra-Empfangssystems im Header der Response im Kindelement `report` des Elements `ResponseDetails` abgelegt.
- `fire-and-forget`: Es wird keine Response erwartet.

Um dem Sender bei dem Empfang einer Response möglichst viele Informationen in Bezug auf den ursprünglichen Request an die Hand zu geben, ist der Response-Header weitgehend eine Kopie des auslösenden Request-Headers, erweitert um das Element `Response-Details`, in dem das eXTra-Empfangssystem u.a. den erfolgreichen Transport oder aufgetretene Fehler übermittelt.

3.3 Versionierung

eXTra definiert eine Menge von XML-Komponenten in unterschiedlichen Namensräumen, die separat entwickelt und gepflegt werden (können) und daher eigene Versionsbezeichner führen. Versionsbezeichner bestehen aus Hauptversionsnummer h, Unterversionsnummer u und Korrekturnummer k und haben die Form h.u[.k]; die Korrekturnummer wird nur verwendet, wenn sie von ihrem impliziten Wert 0 abweicht.

Die Freigabe einer eXTra-Version wird als Release bezeichnet und umfasst eine definierte Menge der XML-Komponenten in je genau einer Version. eXTra-Releases führen einen eigenen Versionsbezeichner.

Die unten stehenden Versionsmatrizen stellen die Beziehungen zwischen der Version eines eXTra-Release und den Versionen der darin enthaltenen XML-Komponenten dar. Die Versionsmatrix Transport definiert eine exklusive Kombination von Release-Version und einer Version jeder XML-Komponente; XML-Komponenten MÜSSEN in genau der angegebenen Kombination von Versionen benutzt werden. Standardnachrichten sind per Definition Nutzdaten und als solche vollkommen transparent für die Transportschicht, und daher ist die Kombination von Release-Version und einer Version einer Standardnachricht nicht exklusiv. Die Versionsmatrix gibt in diesem Fall nur an, ab welchem Release die angegebene Version einer Standardnachricht verfügbar wurde.

Erläuterung der Einträge:

1.0 Komponente in der angegebenen Version verfügbar
~~1.0~~ Komponente in der angegebenen Version verfügbar, aber veraltet (deprecated)
- Komponente nicht verfügbar

Versionsmatrix Transport

Die Komponenten MÜSSEN in der angegebenen Kombination von Versionen verwendet werden.

eXTra Release	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**
Basisstandard					
Request	1.0	1.1	1.2	1.3	1.4
Response					
Logging					
Codelists					
Components					
Error	-	-	1.0	1.0	1.0
Plug-Ins					
Certificates	-	1.0	1.0	1.0	1.0
Communication	1.0	~~1.0~~	-	-	-
Contacts	-	-	1.0	1.0	1.0
DataSource	1.0	1.0	1.0	1.0	1.0
DataTransforms	1.0	1.1	1.2	1.2	1.2

Versionsmatrix Standardnachrichten

Die Komponenten sind in der angegebenen Version ab dem genannten Release verfügbar.

eXTra Standardnachrichten Release	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**
Standardnachrichten					
ConfirmationOfReceipt	1.0	1.1	1.2	1.3	1.3
DataRequest	1.0	1.1	1.1	1.2	1.3
ListOfConfirmationOfReceipt	-	-	1.0	1.1	1.1
ListOfDataRequest	-	-	1.0	1.1	1.1
ListOfStatusResponse	-	-	-	1.0	1.0
ListRequest	-	-	-	-	1.0
ListResponse	-	-	-	-	1.0
StatusRequest	-	-	-	1.0	1.1
StatusResponse	-	-	-	1.0	1.0

4 eXTra Codelisten [normativ]

Die im Folgenden beschriebenen Codelisten sind im Schema-Modul `eXTra-codelists-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 Codeliste CompressionAlgorithmCode

	xcode:CompressionAlgorithmCodeType
Inhalt	Identifikatoren für Komprimierungsalgorithmen
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Ja

Vordefinierte Werte:

```
http://www.extra-standard.de/transforms/compression/NONE
http://www.extra-standard.de/transforms/compression/GZIP
http://www.extra-standard.de/transforms/compression/ZIP
http://www.extra-standard.de/transforms/compression/JET
```

4.2 Codeliste DataContainerCode

	xcode:DataContainerCodeType
Inhalt	Identifikatoren für Containertypen
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Nein

Vordefinierte Werte:

```
http://www.extra-standard.de/container/DATABASE
http://www.extra-standard.de/container/FILE
```

4.3 Codeliste DataTypeCode

	xcode:DataTypeCodeType
Inhalt	Identifikatoren für Datentypen
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Nein

Vordefinierte Werte:

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

4.4 Codeliste EncryptionAlgorithmCode

	xcode:EncryptionAlgorithmCodeType
Inhalt	Identifikatoren für Verschlüsselungsalgorithmen
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Ja

Vordefinierte Werte:

```
http://www.extra-standard.de/transforms/encryption/NONE
http://www.extra-standard.de/transforms/encryption/PKCS7
http://www.extra-standard.de/transforms/encryption/PEM
```

4.5 Codeliste OperationCode

	xcode:OperationCodeType
Inhalt	Identifikatoren für Beschreibung von Operationen im Log
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Nein

Vordefinierte Werte:

```
http://www.extra-standard.de/operation/UNDEFINED
http://www.extra-standard.de/operation/SEND
http://www.extra-standard.de/operation/RECEIVE
http://www.extra-standard.de/operation/ENCRYPT
http://www.extra-standard.de/operation/DECRYPT
http://www.extra-standard.de/operation/COMPRESS
http://www.extra-standard.de/operation/DECOMPRESS
http://www.extra-standard.de/operation/VALIDATE
http://www.extra-standard.de/operation/SIGN
http://www.extra-standard.de/operation/CHECK-SIGNATURE
```

4.6 Codeliste ScenarioCode

	xcode:ScenarioCodeType
Inhalt	Identifikatoren für das Kommunikationszenario
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Ja

Vordefinierte Werte:

```
http://www.extra-standard.de/scenario/fire-and-forget
http://www.extra-standard.de/scenario/request-with-response
http://www.extra-standard.de/scenario/request-with-acknowledgement
```

4.7 Codeliste SignatureAlgorithmCode

	xcode:SignatureAlogrithmType
Inhalt	Identifikatoren für Signaturverfahren
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Ja

Vordefinierte Werte:

```
http://www.extra-standard.de/transforms/signature/NONE
http://www.extra-standard.de/transforms/signature/PKCS7
http://www.extra-standard.de/transforms/signature/PEM
```

4.8 Codeliste TestIndicatorCode

	xcode:TestIndicatorCodeType
Inhalt	Identifikatoren für Erkennung und Verarbeitung von Testfällen
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Nein

Vordefinierte Werte:

```
http://www.extra-standard.de/test/NONE
http://www.extra-standard.de/test/RECEIVE
http://www.extra-standard.de/test/ACCEPT
http://www.extra-standard.de/test/PROCESS
```

4.9 Codeliste UsageCode

	xcode:UsageCodeType
Inhalt	Identifikatoren die Verwendung im Log dokumentierter Parameter
Datentyp	`xsd:anyURI`
Individualisierbar	Ja
Beliebige Domain	Nein

Vordefinierte Werte:

```
http://www.extra-standard.de/usage/UNDEFINED
http://www.extra-standard.de/usage/IN
http://www.extra-standard.de/usage/OUT
http://www.extra-standard.de/usage/INOUT
```

4.10 Codeliste WeightCode

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

Vordefinierte Werte:

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

4.11 Codeliste XSDTypeCodes

	xcode:XSDPrefixedTypeCodes
Inhalt	Liste von XSD Datentypen in präfigierter Form
Datentyp	`xsd:QName`

Vordefinierte Werte: Eine Liste der Namen aller in [XSD] definierten Datentypen mit dem Präfix `xsd`. Beispiele: `xsd:string`, `xsd:QName`.

5 Nachrichtentyp Request [normativ]

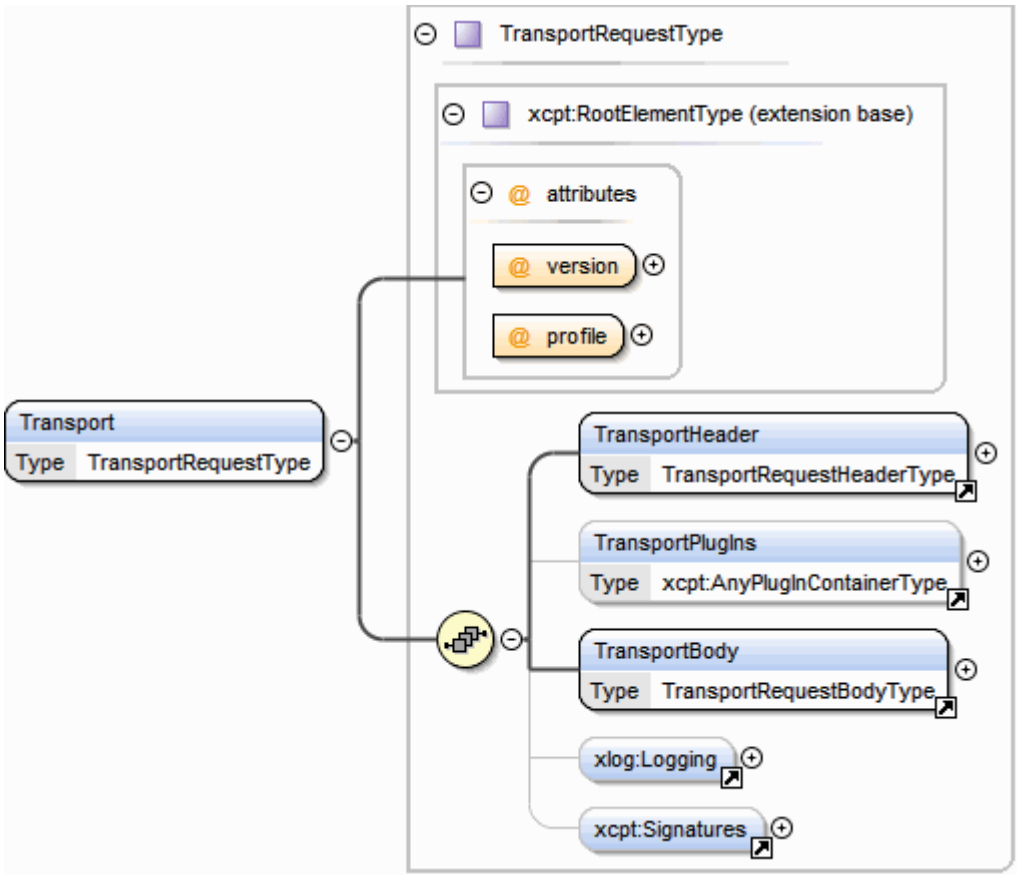
Eine Nachricht vom Typ Request leitet einen Übertragungsvorgang im Sinne des eXTra-Kommunikationsmodells ein. Die Namensraum-URI ist

`http://www.extra-standard.de/namespace/request/1`

Der lokale Name des Wurzelelementes ist `Transport`. Es wird empfohlen, den Namensraumpräfix `xreq` oder alternativ `xtra` zu verwenden.

5.1 Element Transport

	`xreq:Transport`
Inhalt	Enthält eine Instanz der Transportebene eines eXTra Requests
Datentyp	`##elements`
Benutzt von	-



Attribute

	`@profile`	*erforderlich*
Inhalt	Enthält die ID des eXTra Profils	
Datentyp	`xsd:anyURI`	

	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung des eXTra Schemas	
Datentyp	`xsd:anyURI`	

Profilierung

Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
@profile						+
xcpt:Signatures	+	+	+		+	
xlog:Logging	+	+	+		+	
xreq:TransportPlugins	+	+	+		+	

Beschreibung

`xreq:Transport` ist das Wurzelelement einer eXTra-Nachricht vom Typ Request. Das Attribut `profile` enthält die ID des verwendeten eXTra-Profils; diese hat stets die Form einer URI. Das Attribut `version` enthält die Versionsbezeichnung.

Die Elemente `xreq:TransportHeader` und `xreq:TransportBody` sind stets erforderlich und unterliegen nicht der Profilierung.

Beispiel

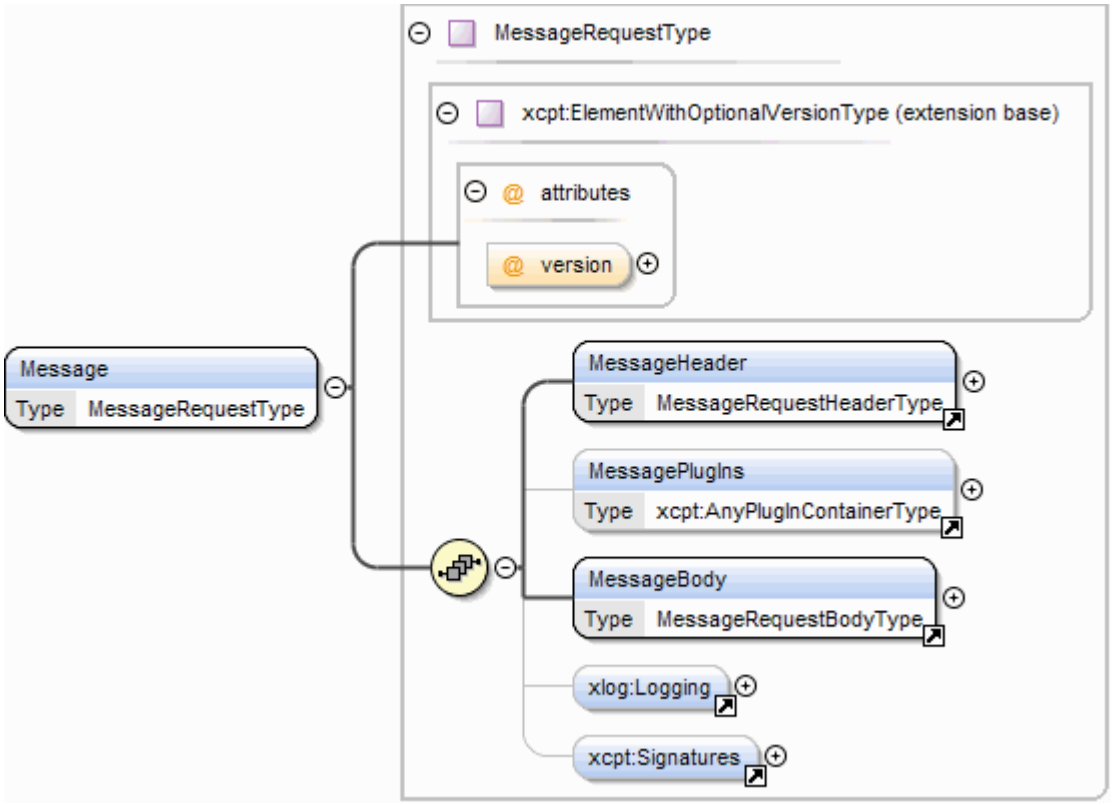
```
<?xml version="1.0" encoding="UTF-8"?>
<xreq:Transport
    xmlns:xreq="http://www.extra-standard.de/namespace/request/1"
    xmlns:xcpt="http://www.extra-standard.de/namespace/components/1"
    profile="http://www.myCompany.de/profile/myProfile/1">
    version="1.4"
    <xreq:TransportHeader>
        <xcpt:TestIndicator>http.//www.extra-standard.de/test/NONE</xcpt:TestIndicator>
        <xcpt:Sender>
            <xcpt:SenderID>12341111</xcpt:SenderID>
        </xcpt:Sender>
       <xcpt:Receiver>
           <xcpt:ReceiverID>55557777</xcpt:ReceiverID>
       </xcpt:Receiver>
        <xcpt:RequestDetails>
            <xcpt:RequestID>1111111</xcpt:RequestID>
            <xcpt:Procedure>http://www.destatis.de/procedures/eCORE#0122</xcpt:Procedure>
            <xcpt:DataType>http://www.destatis.de/datatype/KU101/2008</xcpt:DataType>
            <xcpt:Scenario>http://www.extra-standard.de/scenario/request-with-
acknowledgement</xcpt:Scenario>
        </xcpt:RequestDetails>
    </xreq:TransportHeader>
    <xreq:TransportBody>
        <xcpt:Data>
            <xcpt:AnyXML></xcpt:AnyXML>
        </xcpt:Data>
    </xreq:TransportBody>
</xreq:Transport>
```

5.2 Element XMLTransport

Das Element `xreq:XMLTransport` ist ab Version 1.3.0 durch `xreq:Transport` ersetzt und soll nicht mehr verwendet werden. Es wird nur noch für eine Übergangszeit unterstützt. Die Änderung bezieht sich nur auf den Namen, hinsichtlich aller anderen Eigenschaften sind beide Elemente identisch. Diese Maßnahme stellt eine konsistente Namensgebung der Elemente, welche die Ebenen des Kommunikationsmodells darstellen, her. Außerdem wird ein Verstoß gegen die XML Spezifikation [XML] vermieden, welche mit XML beginnende Namen für eigene Zwecke reserviert.

5.3 Element Message

	`xreq:Message`
Inhalt	Enthält eine Instanz der Nachrichtenebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:PackageBody`, `xreq:TransportBody`,



Attribute

	`@version`	*optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden.*	
Datentyp	`xsd:anyURI`	

Profilierung

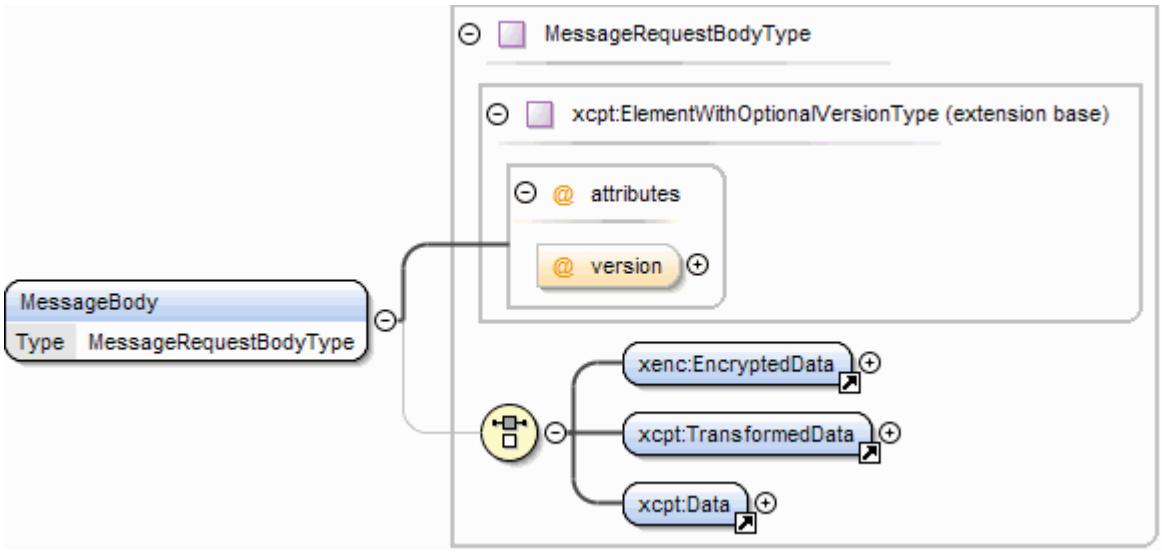
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`xcpt:Signatures`	+	+	+		+	
`xlog:Logging`	+	+	+		+	
`xreq:MessagePlugins`	+	+	+		+	

Beschreibung

`xreq:Message` repräsentiert eine Instanz der Nachrichtenebene eines eXTra-Requests und enthält mindestens die Elemente `xreq:MessageHeader` und `xreq:MessageBody`.

5.4 Element MessageBody

	`xreq:MessageBody`
Inhalt	Enthält den Body einer Instanz der Nachrichtenebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:Message`



Attribute

	`@version`	*optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.	
Datentyp	`xsd:anyURI`	

Profilierung

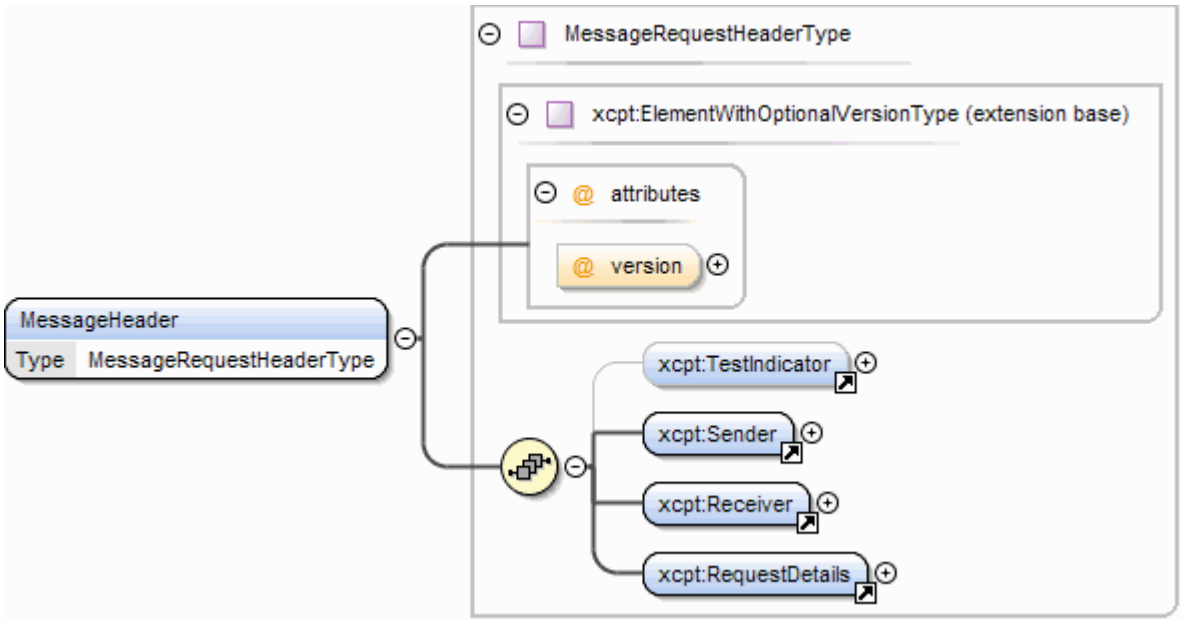
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`xenc:EncryptedData`	+	+	+			
`xcpt:TransformedData`	+	+	+			
`xcpt:Data`			*IMPLICIT*			

Beschreibung

`xreq:MessageBody` enthält den Body einer Instanz der Nachrichtenebene eines eXTra-Requests. Im Body sind die Nutzdaten in verschlüsselter oder unverschlüsselter Form abgelegt.

5.5 Element MessageHeader

	`xreq:MessageHeader`
Inhalt	Enthält den Header einer Instanz der Nachrichtenebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:Message`



Attribute

	`@version`	*optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.	
Datentyp	`xsd:anyURI`	

Profilierung

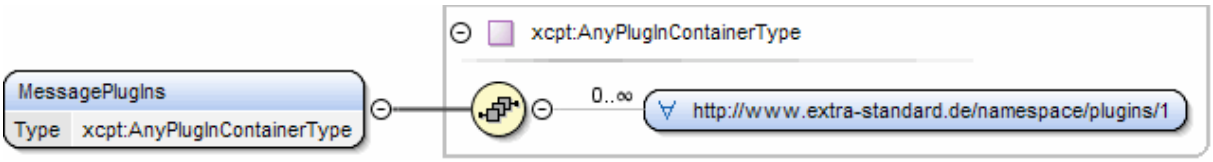
Der Inhalt des Elementes unterliegt nicht der Profilierung.

Beschreibung

Das Element `xreq:MessageHeader` enthält den Header einer Instanz der Nachrichtenebene eines eXTra-Requests.

5.6 Element MessagePlugIns

	xreq:MessagePlugIns
Inhalt	Enthält PlugIn-Strukturen der Nachrichtenebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:Message`



Profilierung

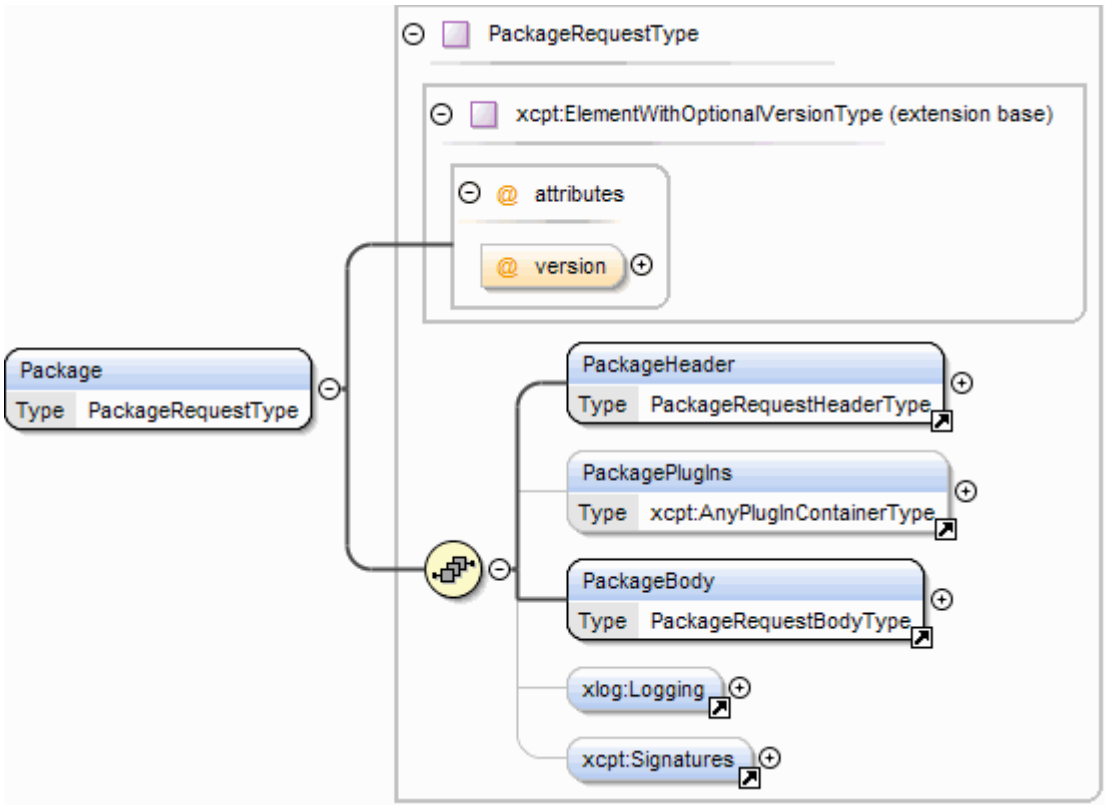
Verwendung und Inhalt des Elementes erfordern eine Profilierung. Es darf nur PlugIn-Elemente enthalten, die per Profilierung als zulässig oder erforderlich deklariert wurden.

Beschreibung

`xreq:MessagePlugIns` enthält PlugIn-Strukturen der Nachrichtenebene eines eXTra-Requests.

5.7 Element Package

	xreq:Package
Inhalt	Enthält eine Instanz der Paketebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:TransportBody`



Attribute

	@version *optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.
Datentyp	`xsd:anyURI`

Profilierung

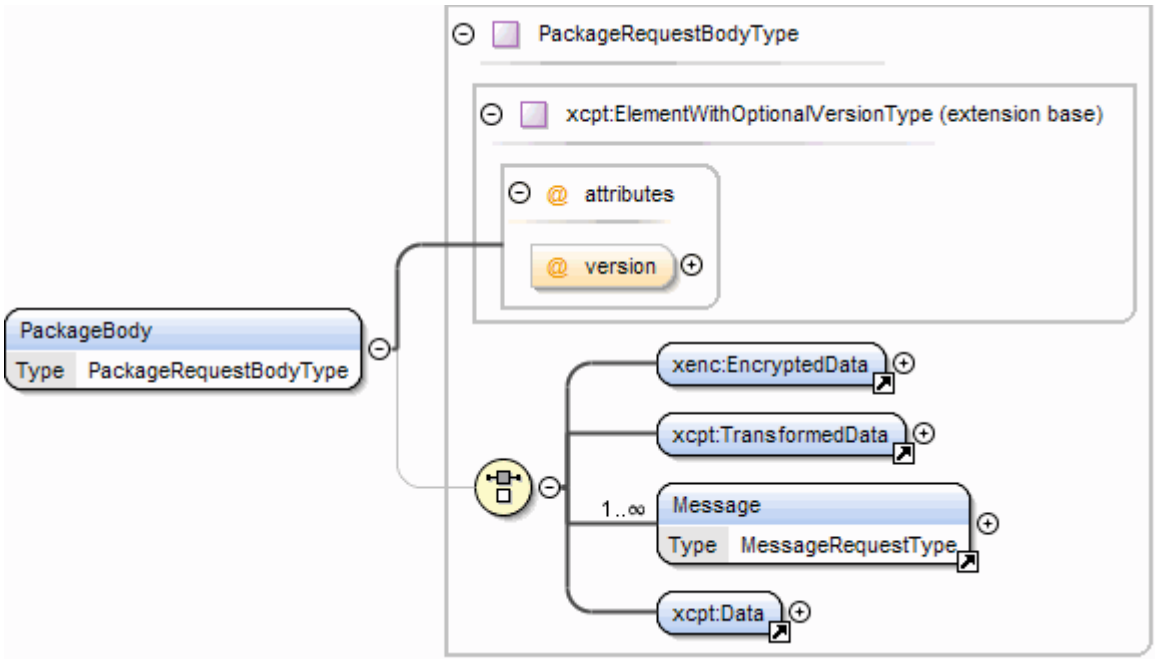
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
xcpt:Signatures	+	+	+		+	
xlog:Logging	+	+	+		+	
xreq:PackagePlugins	+	+	+		+	

Beschreibung

`xreq:Package` enthält eine Instanz der Paketebene eines eXTra-Requests.

5.8 Element PackageBody

	`xreq:PackageBody`
Inhalt	Enthält den Body einer Instanz der Paketebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:Package`



Attribute

	`@version`	*Optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.	
Datentyp	`xsd:anyURI`	

Profilierung

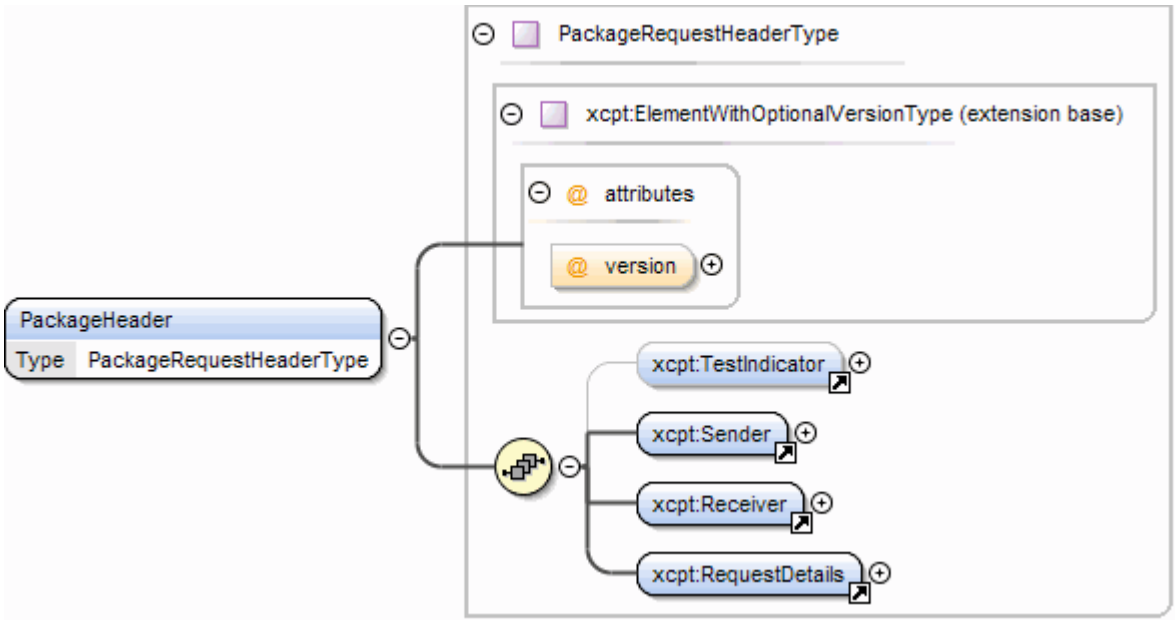
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`xenc:EncryptedData`	+	+	+			
`xcpt:TransformedData`	+	+	+			
`xcpt:Data`			*IMPLICIT*			

Beschreibung

`xreq:PackageBody` enthält den Body einer Instanz der Paketebene eines eXTra-Requests. Dieser enthält – in verschlüsselter oder unverschlüsselter Form – entweder eine Instanz der Nachrichtenebene (`xreq:Message`) oder unmittelbar eine Nutzdateninstanz (`xcpt:Data`).

5.9 Element PackageHeader

	`xreq:PackageHeader`
Inhalt	Enthält den Header einer Instanz der Paketebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:Package`



Attribute

	`@version`	*Optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.	
Datentyp	`xsd:anyURI`	

Profilierung

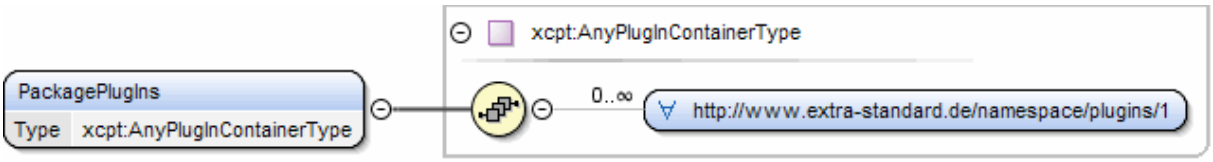
Der Inhalt des Elementes unterliegt nicht der Profilierung.

Beschreibung

`xreq:PackageHeader` enthält den Header einer Instanz der Paketebene eines eXTra-Requests.

5.10 Element PackagePlugIns

	xreq:PackagePlugIns
Inhalt	Enthält PlugIn-Strukturen der Paketebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:Package`



Profilierung

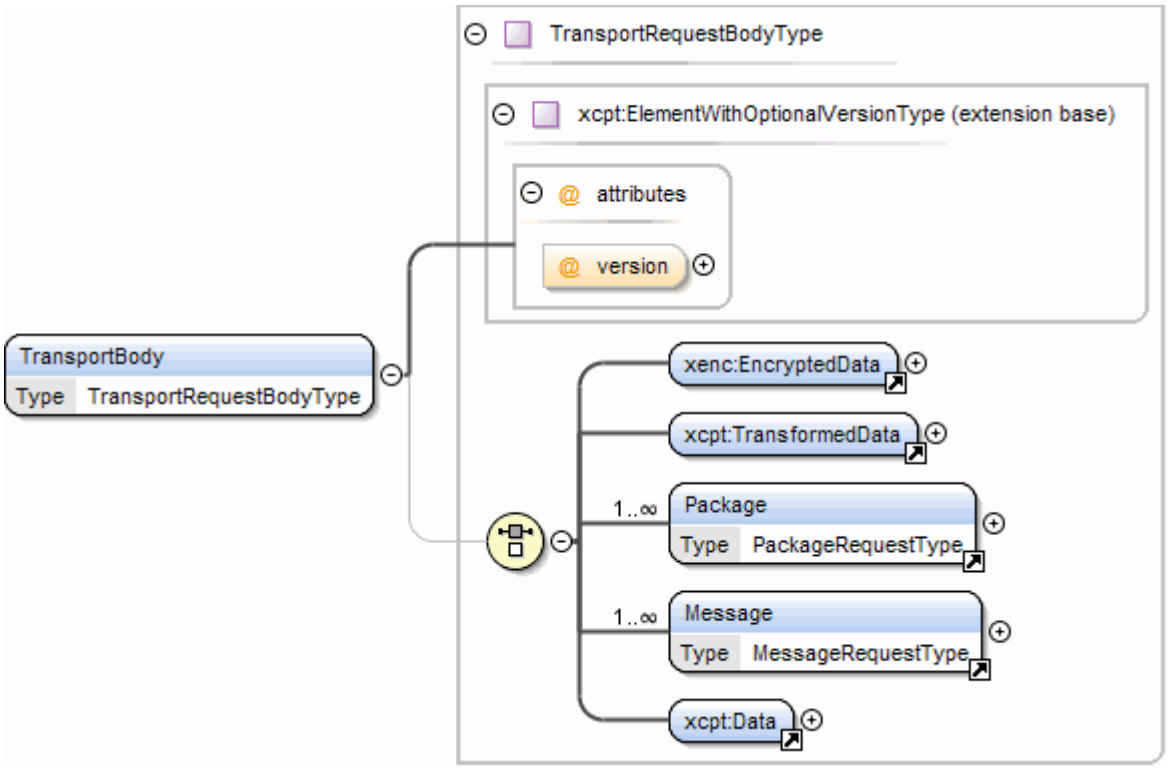
Verwendung und Inhalt des Elementes erfordern eine Profilierung. Es darf nur PlugIn-Elemente enthalten, die per Profilierung als zulässig oder erforderlich deklariert wurden.

Beschreibung

`xreq:PackagePlugIns` enthält PlugIn-Strukturen der Paketebene eines eXTra-Requests.

5.11 Element TransportBody

	`xreq:TransportBody`
Inhalt	Enthält den Body der Transportebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:XMLTransport`



Attribute

	`@version` *Optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.
Datentyp	`xsd:anyURI`

Profilierung

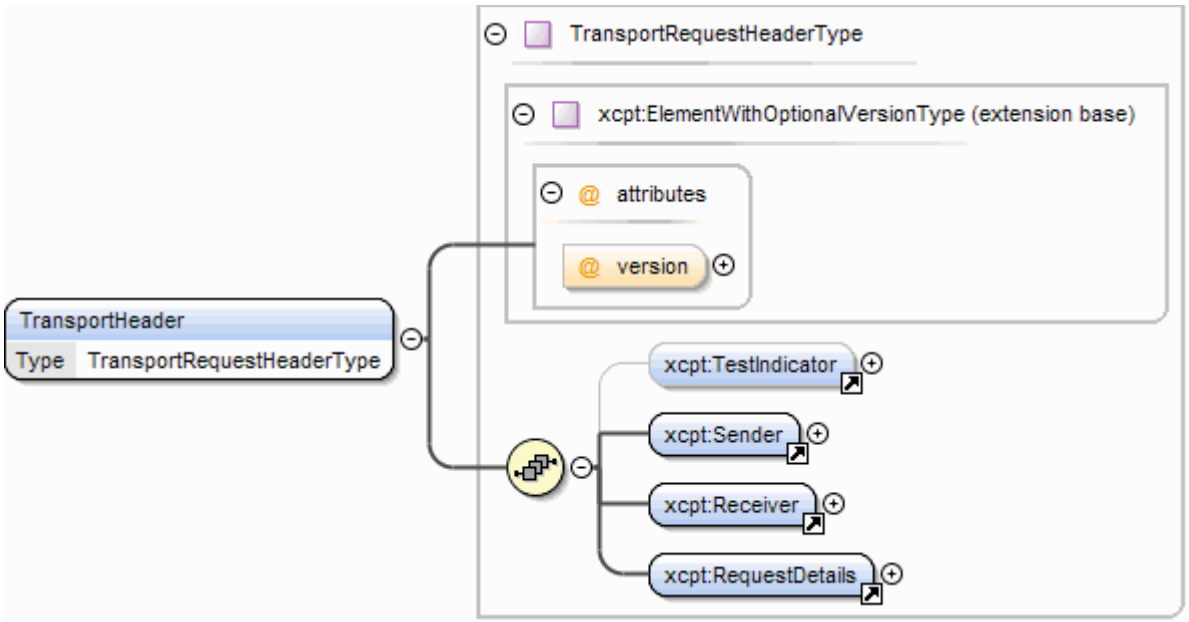
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`xenc:EncryptedData`	+	+	+			
`xcpt:TransformedData`	+	+	+			
`xreq:Message`	+	+	+	+	+	
`xreq:Package`	+	+	+	+	+	
`xreq:Data`			*IMPLICIT*			

Beschreibung

`xreq:TransportBody` enthält den Body der Transportebene eines eXTra-Requests. In diesem befindet sich entweder eine Folge von Paketen (`xreq:Package`) oder eine Folge von Nachrichten (`xreq:Message`) oder ein Nutzdatenblock (`xcpt:Data`). Der Inhalt kann verschlüsselt sein; in diesem Fall enthält der Body nur ein Element `xcpt:TransformedData` oder `xenc:EncryptedData`. Sämtliche Kindelemente unterliegen der Profilierung; da sie sich gegenseitig ausschließen, bestehen hierbei Wechselwirkungen. Das Vorkommen von `xcpt:Data` wird implizit durch die Profilierung von `xreq:Message` und `xreq:Package` gesteuert; siehe [PROF].

5.12 Element TransportHeader

	`xreq:TransportHeader`
Inhalt	Enthält den Header der Transportebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:XMLTransport`



Attribute

	`@version` *optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.
Datentyp	`xsd:anyURI`

Profilierung

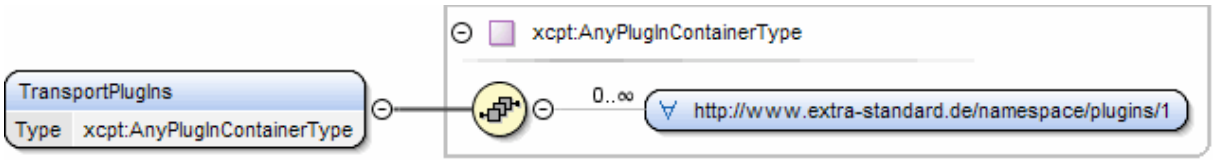
Der Inhalt des Elementes unterliegt nicht der Profilierung.

Beschreibung

`xreq:TransportHeader` enthält den Header der Transportebene eines eXTra-Requests.

5.13 Element TransportPlugIns

	`xreq:TransportPlugIns`
Inhalt	Enthält PlugIn-Strukturen der Transportebene eines eXTra-Requests
Datentyp	`##elements`
Benutzt von	`xreq:XMLTransport`



Profilierung

Verwendung und Inhalt des Elementes erfordern eine Profilierung. Es darf nur PlugIn-Elemente enthalten, die per Profilierung als zulässig oder erforderlich deklariert wurden.

Beschreibung

`xreq:TransportPlugIns` enthält PlugIn-Strukturen der Transportebene eines eXTra-Requests.

6 Nachrichtentyp Response [normativ]

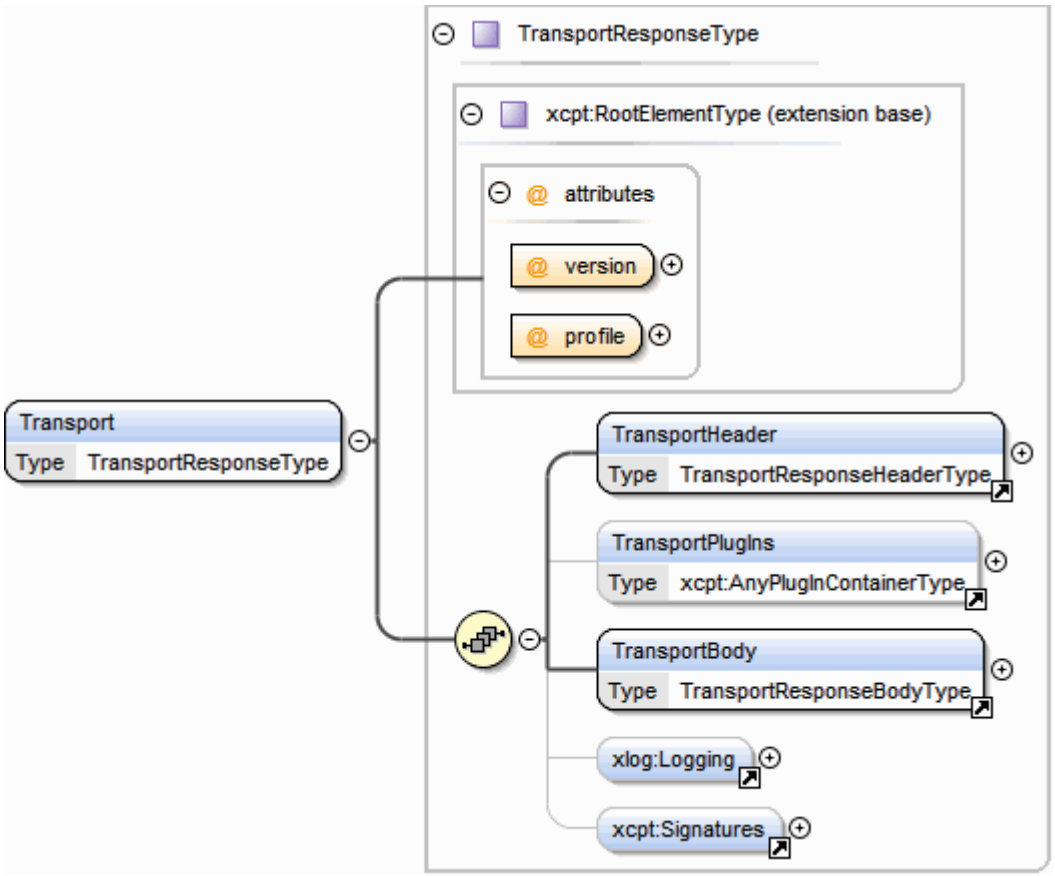
Eine Nachricht vom Typ Response beendet einen Übertragungsvorgang mit einer Antwortnachricht. Die Namensraum-URI ist

http://www.extra-standard.de/namespace/response/1

Der lokale Name des Wurzelelementes ist `Transport`. Es wird empfohlen, den Namensraumpräfix `xres` oder alternativ `xtra` zu verwenden.

6.1 Element Transport

	xres:Transport
Inhalt	Enthält eine Instanz der Transportebene eines eXTra-Response
Datentyp	`##elements`
Benutzt von	-



Attribute

	@profile	*erforderlich*
Inhalt	Enthält die ID des eXTra-Profils	
Datentyp	`xsd:anyURI`	

	@version	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas	
Datentyp	`xsd:anyURI`	

Profilierung

Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
@profile						+
xcpt:Signatures	+	+	+		+	
xlog:Logging	+	+	+		+	
xres:TransportPlugins	+	+	+		+	

Beschreibung

`xres:Transport` ist das Wurzelelement einer eXTra-Nachricht vom Typ Response. Das Attribut `profile` enthält die ID des verwendeten eXTra-Profils; diese hat stets die Form einer URI. Das Attribut `version` enthält die Versionsbezeichnung.

Die Elemente `xres:TransportHeader` und `xres:TransportBody` sind stets erforderlich und unterliegen nicht der Profilierung.

Beispiel

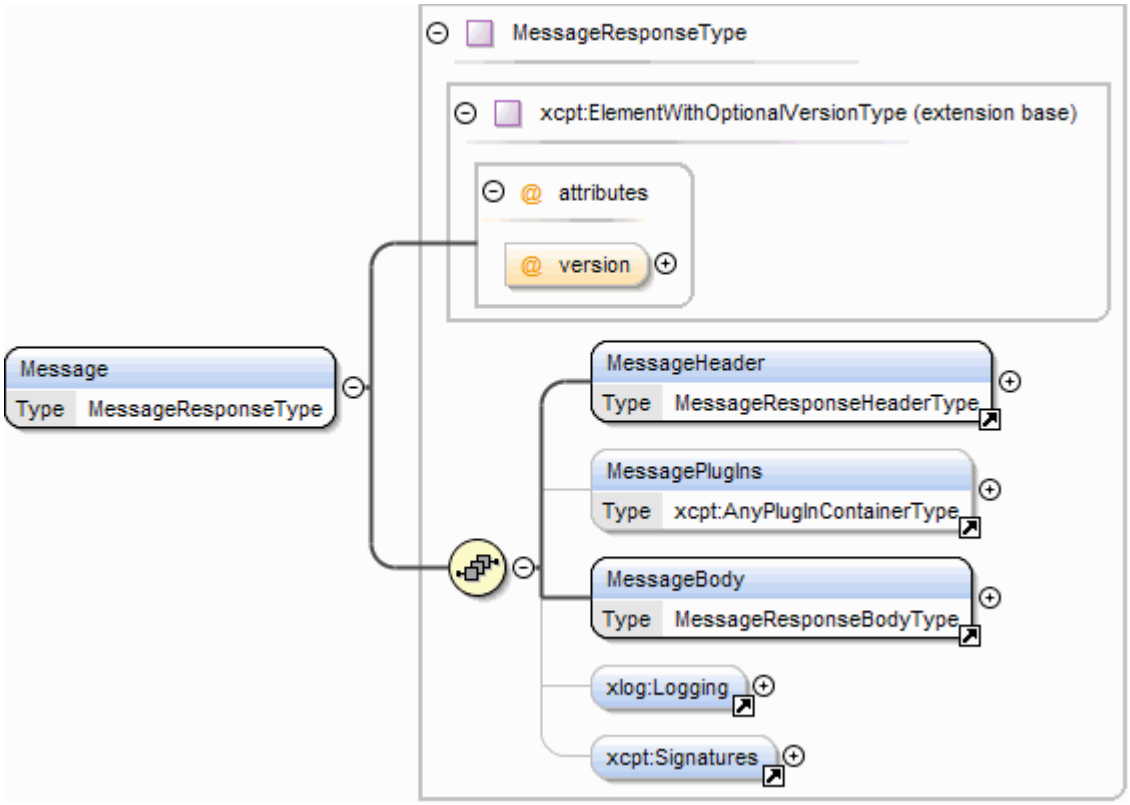
```
<?xml version="1.0" encoding="UTF-8"?>
<xres:Transport
    xmlns:xres="http://www.extra-standard.de/namespace/response/1"
    xmlns:xcpt="http://www.extra-standard.de/namespace/components/1"
    profile="http://www.myCompany.de/profile/myProfile/1">
    version="1.4"
    <xres:TransportHeader>
        <xcpt:TestIndicator>http.//www.extra-standard.de/test/NONE</xcpt:TestIndicator>
        <xcpt:Sender>
            <xcpt:SenderID>12341111</xcpt:SenderID>
        </xcpt:Sender>
       <xcpt:Receiver>
           <xcpt:ReceiverID>55557777</xcpt:ReceiverID>
       </xcpt:Receiver>
        <xcpt:RequestDetails>
            <xcpt:RequestID>1111111</xcpt:RequestID>
            <xcpt:Procedure>http://www.destatis.de/procedures/eCORE#0122</xcpt:Procedure>
            <xcpt:DataType>http://www.destatis.de/datatype/KU101/2008</xcpt:DataType>
            <xcpt:Scenario>http://www.extra-standard.de/scenario/request-with-
acknowledgement</xcpt:Scenario>
        </xcpt:RequestDetails>
        <xcpt:ResponsestDetails>
            <xcpt:ResponseID>201309260609261111</xcpt:ResponseID>
            <xcpt:TimeStamp>2013-09-26T06:09:26</xcpt:TimeStamp>
            <xcpt:Report highestWeight="http://www.extra-standard.de/weight/INFO">
                  <xcpt:Flag weight="http://www.extra-standard.de/weight/INFO">
                        <xcpt:Code>T1000</xcpt:Code>
                        <xcpt:Text>Transport Ebene:alles ok</xcpt:Text>
                  </xcpt:Flag>
            </xcpt:Report>
        </xcpt:ResponseDetails>
    </xres:TransportHeader>
    <xres:TransportBody/>
</xres:Transport>
```

6.2 Element XMLTransport

Das Element `xres:XMLTransport` ist ab Version 1.3.0 durch `xres:Transport` ersetzt und soll nicht mehr verwendet werden. Es wird nur noch für eine Übergangszeit unterstützt. Die Änderung bezieht sich nur auf den Namen, hinsichtlich aller anderen Eigenschaften sind beide Elemente identisch. Diese Maßnahme stellt eine konsistente Namensgebung der Elemente, welche die Ebenen des Kommunikationsmodells darstellen, her. Außerdem wird ein Verstoß gegen die XML Spezifikation [XML] vermieden, welche mit XML beginnende Namen für eigene Zwecke reserviert.

6.3 Element Message

	`xres:Message`
Inhalt	Enthält eine Instanz der Nachrichtenebene eines eXTra-Response
Datentyp	`##elements`
Benutzt von	`xres:PackageBody`, `xres:TransportBody`,



Attribute

	`@version`	*optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden.*	
Datentyp	`xsd:anyURI`	

Profilierung

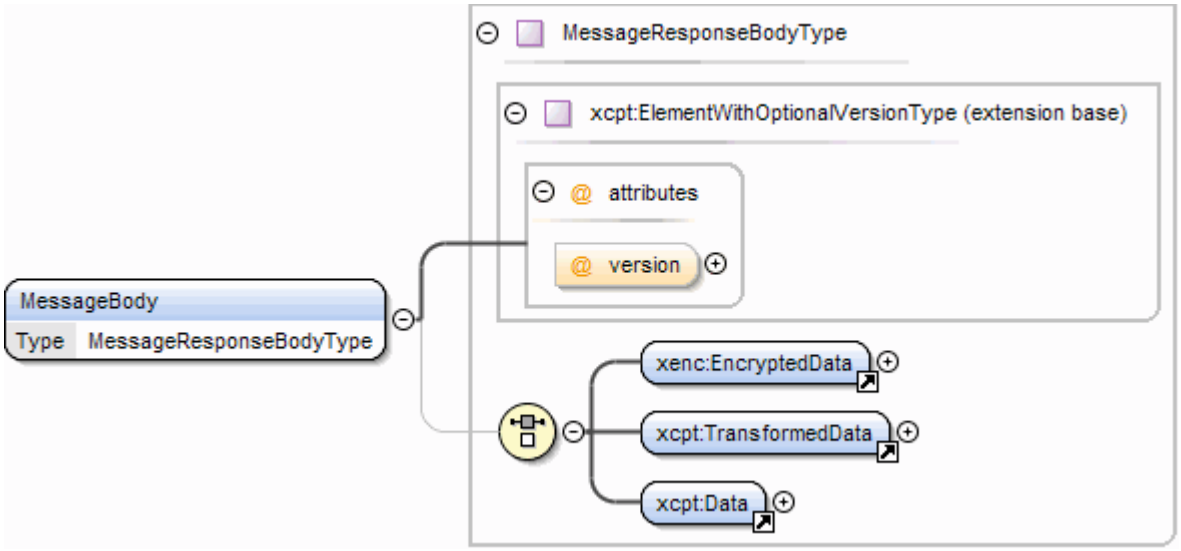
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`xcpt:Signatures`	+	+	+		+	
`xlog:Logging`	+	+	+		+	
`xres:MessagePlugins`	+	+	+		+	

Beschreibung

`xres:Message` repräsentiert eine Instanz der Nachrichtenebene eines eXTra-Response und enthält mindestens die Elemente `xres:MessageHeader` und `xres:MessageBody`.

6.4 Element MessageBody

	`xres:MessageBody`
Inhalt	Enthält den Body einer Instanz der Nachrichtenebene eines eXTra-Response
Datentyp	`##elements`
Benutzt von	`xres:Message`



Attribute

	`@version`	*optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.	
Datentyp	`xsd:anyURI`	

Profilierung

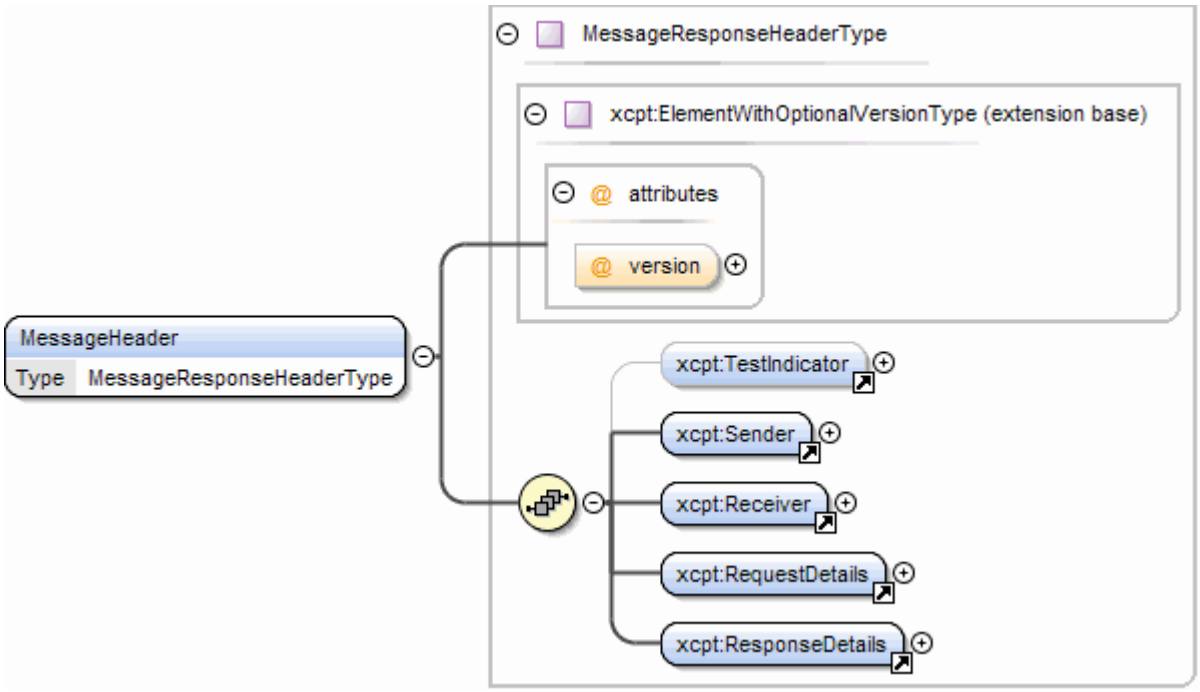
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`xenc:EncryptedData`	+	+	+			
`xlog:Data`			*IMPLICIT*			

Beschreibung

`xres:MessageBody` enthält den Body einer Instanz der Nachrichtenebene eines eXTra-Response. Im Body sind die Nutzdaten in verschlüsselter oder unverschlüsselter Form abgelegt.

6.5 Element MessageHeader

	`xres:MessageHeader`
Inhalt	Enthält den Header einer Instanz der Nachrichtenebene eines eXTra-Response
Datentyp	`##elements`
Benutzt von	`xres:Message`



Attribute

	`@version` *optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.
Datentyp	`xsd:anyURI`

Profilierung

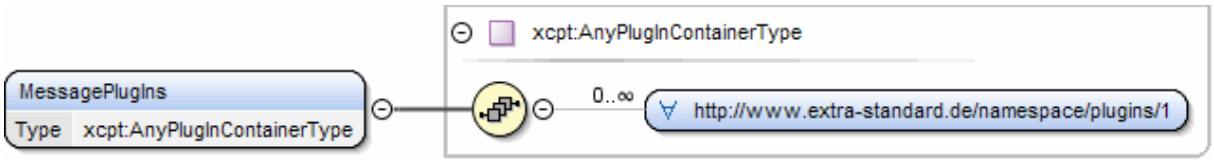
Der Inhalt des Elementes unterliegt nicht der Profilierung.

Beschreibung

`xres:MessageHeader` enthält den Header einer Instanz der Nachrichtenebene eines eXTra-Response.

6.6 Element MessagePlugIns

	`xres:MessagePlugIns`
Inhalt	Enthält PlugIn-Strukturen der Nachrichtenebene eines eXTra Response
Datentyp	`##elements`
Benutzt von	`xres:Message`



Profilierung

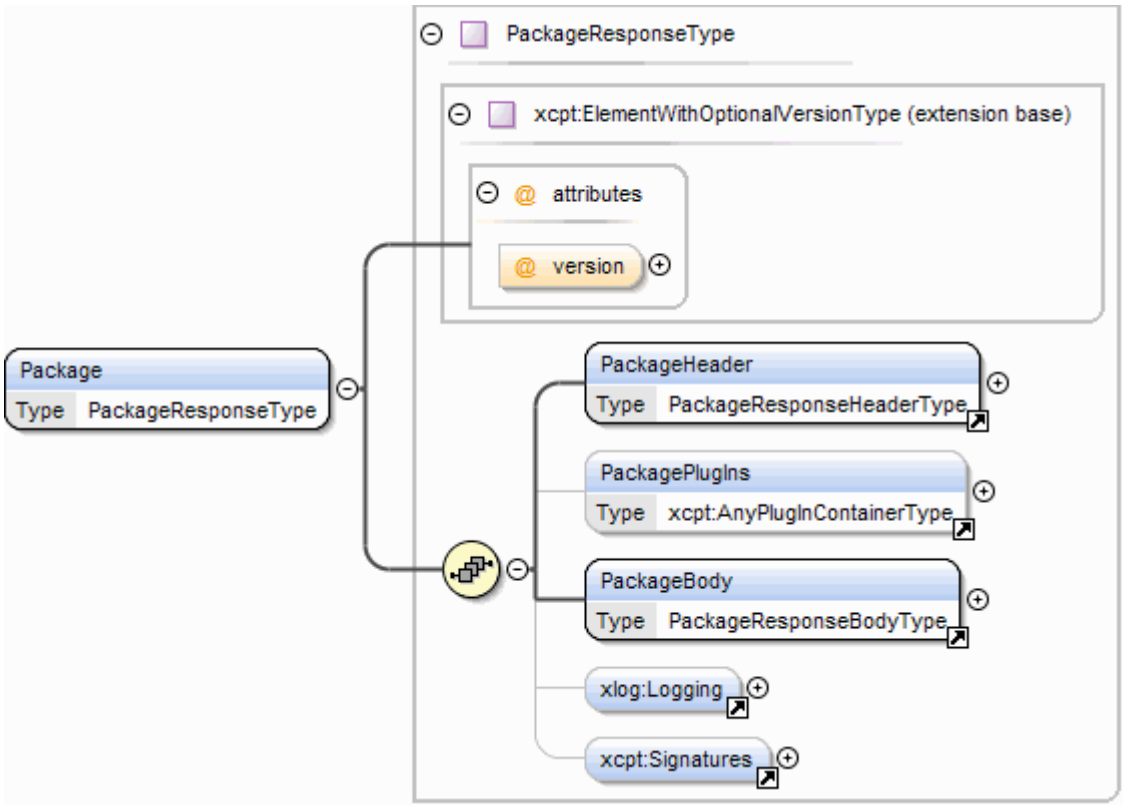
Verwendung und Inhalt des Elementes erfordern eine Profilierung. Es darf nur PlugIn-Elemente enthalten, die per Profilierung als zulässig oder erforderlich deklariert wurden.

Beschreibung

`xres:MessagePlugIns` enthält PlugIn-Strukturen der Nachrichtenebene eines eXTra-Response.

6.7 Element Package

	`xres:Package`
Inhalt	Enthält eine Instanz der Paketebene eines eXTra-Response
Datentyp	`##elements`
Benutzt von	`xres:TransportBody`



Attribute

	`@version` *optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.
Datentyp	`xsd:anyURI`

Profilierung

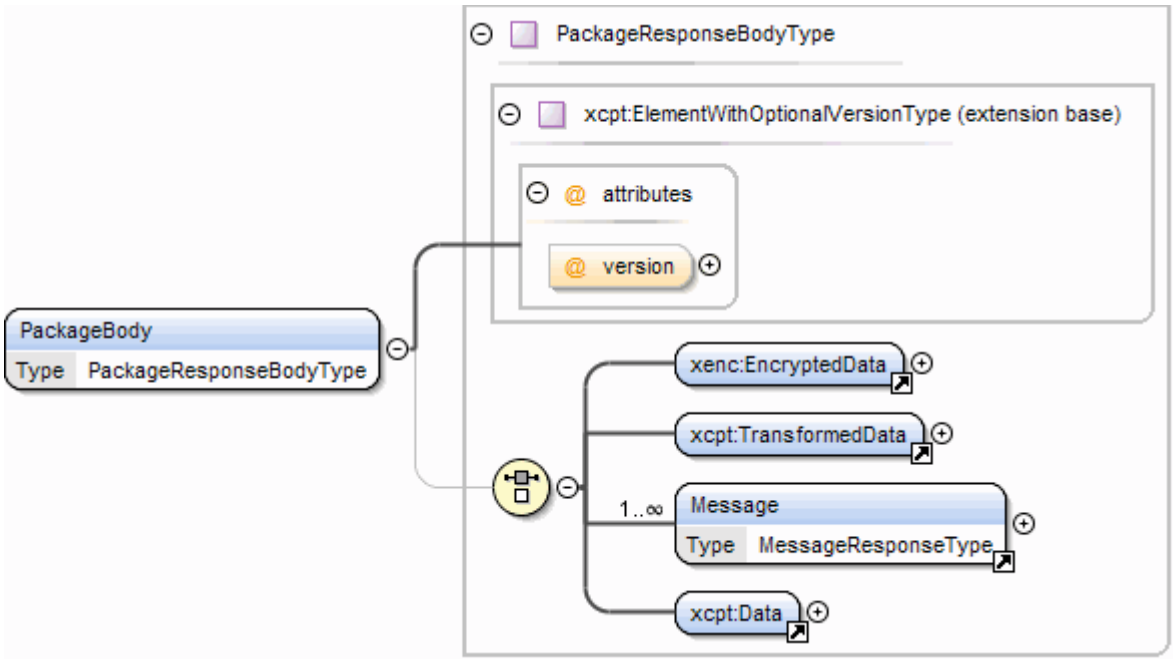
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`xcpt:Signatures`	+	+	+		+	
`xlog:Logging`	+	+	+		+	
`xres:PackagePlugins`	+	+	+		+	

Beschreibung

`xres:Package` enthält eine Instanz der Paketebene eines eXTra-Response.

6.8 Element PackageBody

	xres:PackageBody
Inhalt	Enthält den Body einer Instanz der Paketebene eines eXTra-Response
Datentyp	`##elements`
Benutzt von	`xres:Package`



Attribute

	@version	*optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.	
Datentyp	`xsd:anyURI`	

Profilierung

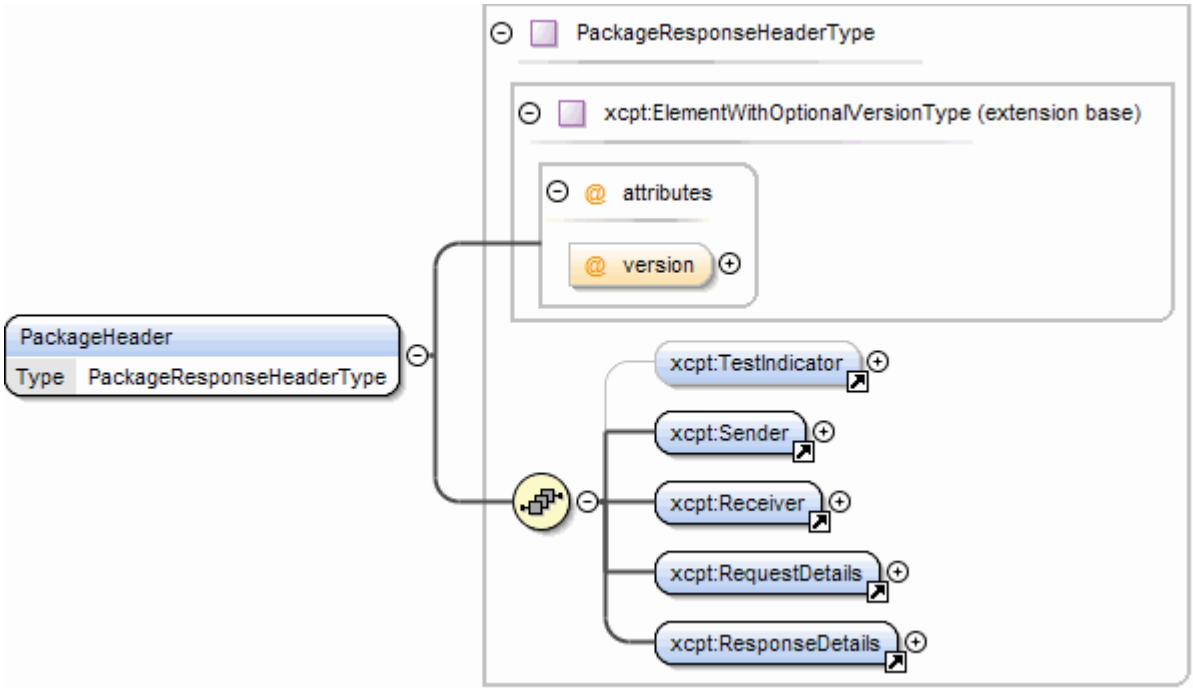
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
xenc:EncryptedData	+	+	+			
xlog:Data			*IMPLICIT*			

Beschreibung

`xres:PackageBody` enthält den Body einer Instanz der Paketebene eines eXTra-Response. Dieser enthält – in verschlüsselter oder unverschlüsselter Form – entweder eine Instanz der Nachrichtenebene (`xres:Message`) oder unmittelbar eine Nutzdateninstanz (`xcpt:Data`).

6.9 Element PackageHeader

	`xres:PackageHeader`
Inhalt	Enthält den Header einer Instanz der Paketebene eines eXTra-Response
Datentyp	`##elements`
Benutzt von	`xres:Package`



Attribute

	`@version`	*optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.	
Datentyp	`xsd:anyURI`	

Profilierung

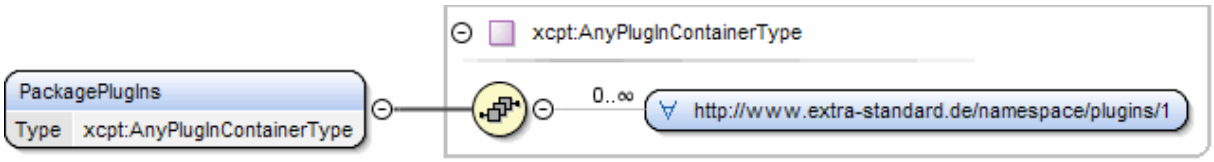
Der Inhalt des Elementes unterliegt nicht der Profilierung.

Beschreibung

`xres:PackageHeader` enthält den Header einer Instanz der Paketebene eines eXTra-Response.

6.10 Element PackagePlugIns

	xres:PackagePlugIns
Inhalt	Enthält PlugIn-Strukturen der Paketebene eines eXTra-Response
Datentyp	`##elements`
Benutzt von	`xres:Package`



Profilierung

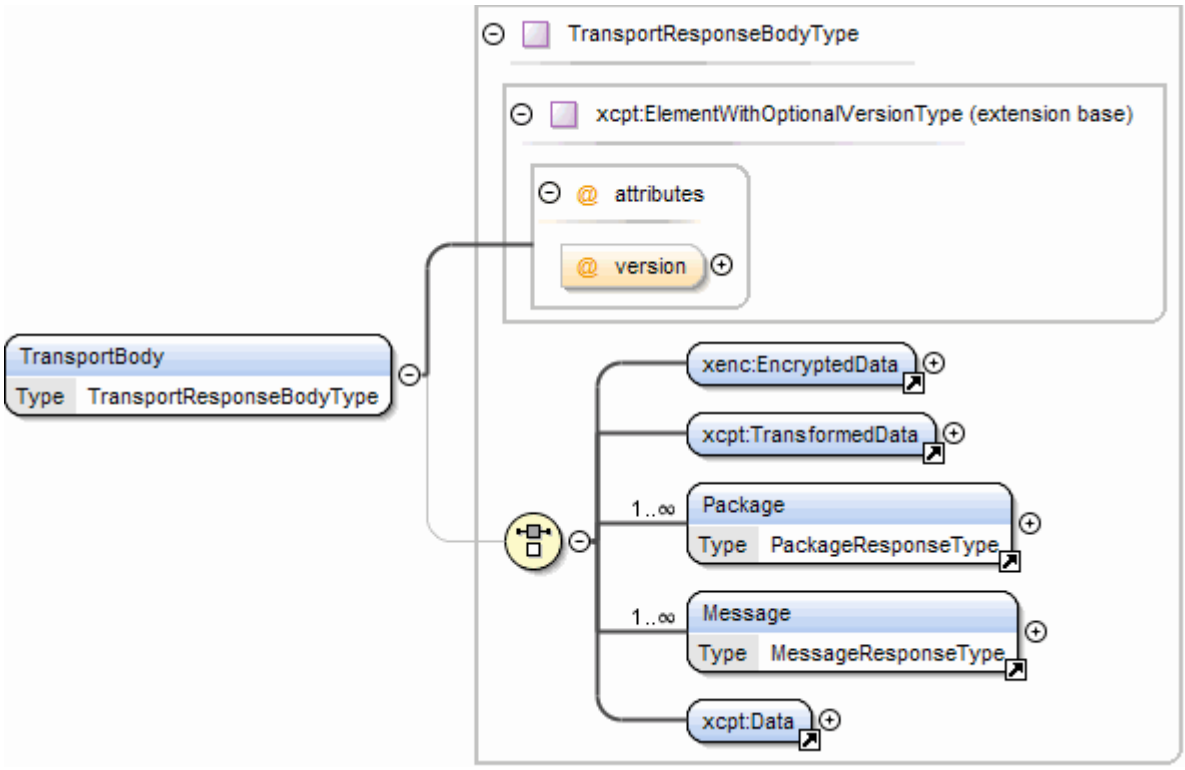
Verwendung und Inhalt des Elementes erfordern eine Profilierung. Es darf nur PlugIn-Elemente enthalten, die per Profilierung als zulässig oder erforderlich deklariert wurden.

Beschreibung

`xres:PackagePlugIns` enthält PlugIn-Strukturen der Paketebene eines eXTra-Response.

6.11 Element TransportBody

	`xres:TransportBody`
Inhalt	Enthält den Body der Transportebene eines eXTra-Response
Datentyp	`##elements`
Benutzt von	`xres:XMLTransport`



Attribute

	`@version` *optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.
Datentyp	`xsd:anyURI`

Profilierung

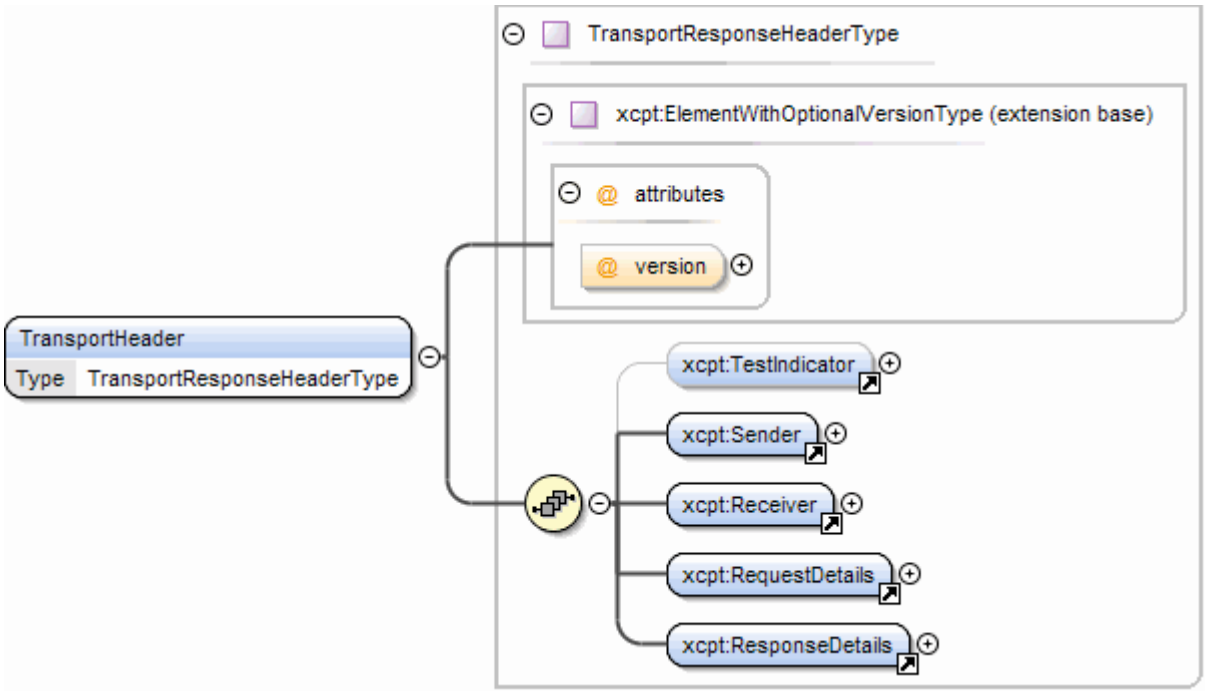
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`xenc:EncryptedData`	+	+	+			
`xlog:Message`	+	+	+	+	+	
`xres:Package`	+	+	+	+	+	
`xres:Data`			*IMPLICIT*			

Beschreibung

`xres:TransportBody` enthält den Body der Transportebene eines eXTra-Response. In diesem befindet sich entweder eine Folge von Paketen (`xres:Package`) oder eine Folge von Nachrichten (`xres:Message`) oder ein Nutzdatenblock (`xcpt:Data`). Der Inhalt kann verschlüsselt sein; in diesem Fall enthält der Body nur ein Element `xenc:EncryptedData`. Sämtliche Kindelemente unterliegen der Profilierung; da sie sich gegenseitig ausschließen, bestehen hierbei Wechselwirkungen. Das Vorkommen von `xcpt:Data` wird implizit durch die Profilierung von `xres:Message` und `xres:Package` gesteuert; siehe [PROF].

6.12 Element TransportHeader

	`xres:TransportHeader`
Inhalt	Enthält den Header der Transportebene eines eXTra Response
Datentyp	`##elements`
Benutzt von	`xres:XMLTransport`



Attribute

	`@version`	*optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.	
Datentyp	`xsd:anyURI`	

Profilierung

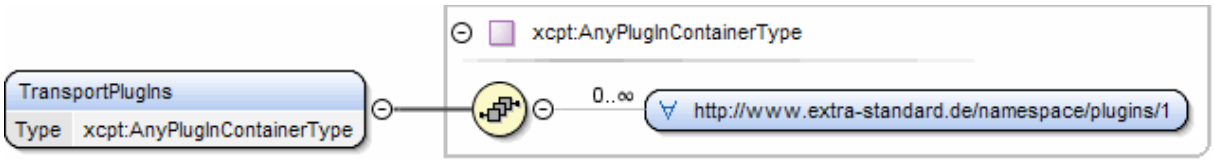
Der Inhalt des Elementes unterliegt nicht der Profilierung.

Beschreibung

`xres:TransportHeader` enthält den Header der Transportebene eines eXTra-Response.

6.13 Element TransportPlugIns

	xres:TransportPlugIns
Inhalt	Enthält PlugIn-Strukturen der Transportebene eines eXTra-Response
Datentyp	`##elements`
Benutzt von	`xres:XMLTransport`



Profilierung

Verwendung und Inhalt des Elementes erfordern eine Profilierung. Es darf nur PlugIn-Elemente enthalten, die per Profilierung als zulässig oder erforderlich deklariert wurden.

Beschreibung

`xres:TransportPlugIns` enthält PlugIn-Strukturen der Transportebene eines eXTra-Response.

7 eXTra-Error [normativ]

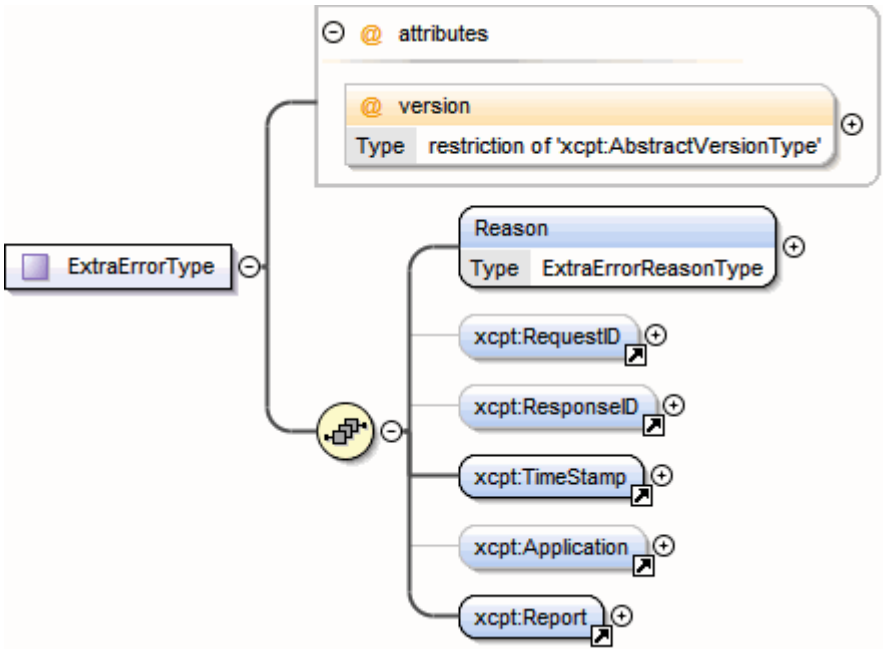
eXTra-Error ist eine spezielle Quittungsnachricht, deren Zweck die Darstellung so schwerwiegender Fehler auf Ebene des eXTra-Protokolls ist, dass Requests weder als akzeptiert noch als verarbeitet gelten können. Tatsache und Details der Verwendung dieser Nachricht sind verfahrensspezifisch zu regeln. Die Namensraum-URI ist

`http://www.extra-standard.de/namespace/service/1`

Der lokale Name des Wurzelelementes ist `eXTra-Error`. Es wird empfohlen, den Namensraumpräfix `xsrv` zu verwenden.

7.1 Element ExtraError

	`xsrv:ExtraError`
Inhalt	Enthält Details zum Fehlerfall
Datentyp	`##elements`
Benutzt von	–



Attribute

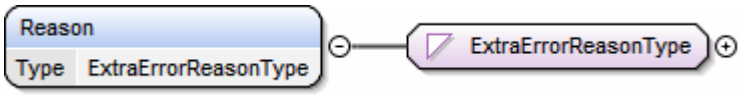
	`@version`	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Error Schemas	
Datentyp	`xsd:anyURI`	

Beschreibung

Enthält eine Beschreibung des Fehlers, der zum Abweisen des Requests führte. Das Element `xsrv:Reason` enthält eine URI, die den Fehler klassifiziert.

7.2 Element Reason

	xsrv:Reason
Inhalt	Klassifiziert die Fehlerursache
Datentyp	`xsd:anyURI`
Benutzt von	–



Beschreibung

Klassifiziert die Fehlerursache mittels folgender vordefinierter URIs:

http://www.extra-standard.de/reason/SERVICE_TEMPORARILY_UNAVAILABLE
http://www.extra-standard.de/reason/SERVICE_PERMANENTLY_UNAVAILABLE
http://www.extra-standard.de/reason/SERVICE_FAILURE
http://www.extra-standard.de/reason/INVALID_REQUEST
http://www.extra-standard.de/reason/UNSPECIFIED

8 Plug-Ins [normativ]

Plug-Ins ergänzen den Kernbereich des Standards um einen profilierbaren Satz von Funktionalitäten. Plug-Ins sind verfahrensunabhängig. Die Namensraum-URI ist

http://www.extra-standard.de/namespace/plugins/1

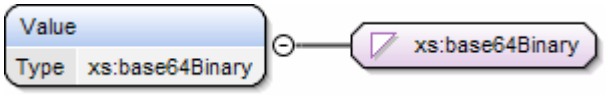
Jedes Plug-In hat ein eigenes lokales Wurzelelement. Der Geltungsbereich eines Plug-In ist analog der dem Kernbereich zugeordneten Funktionalitäten ebenenspezifisch, das Konzept der Trennung der Ebenen wird strikt umgesetzt. Jede Ebene daher verfügt über ein spezifisches, standardmäßig optionales Element `xxxPlugIns`, in dem Plug-Ins abzulegen sind.

eXTra stellt folgende Plug-Ins bereit:

Wurzelelement	*Beschreibung*
`Certificates`	Beschreibt Zertifikate, z.B. für die Verschlüsselung asynchroner Nachrichten an den ursprünglichen Absender.
`Communication`	Dieses Plugin wurde in Version 1.1 durch `Certificates` ersetzt und entfällt vollständig ab Version 1.2.
`Contacts`	Ermöglicht die Angabe von Kontaktdaten.
`DataSource`	Beschreibt in abstrakter Form die Quelle, aus der Nutzdaten stammen. Es kann sich dabei z.B. um eine Datei oder eine Datenbank handeln.
`DataTransforms`	Beschreibt Transformationen wie Verschlüsselung und Komprimierung, sowie die Signierung von Dokumentfragmenten. In Verbindung mit `xcpt:TransformedData` erlaubt es die genauer Dokumentation von Art und Reihenfolge aller Transformationen, die an den Ursprungsdaten vorgenommen wurden.

8.1 Element Value

	`xplg:Value`
Inhalt	Enthält den Schlüssel eines Zertifikates in Base64-Kodierung
Datentyp	`xs:base64Binary`
Benutzt von	`xplg:X509EncCertificate`, `xplg:X509ReplyCertificate`

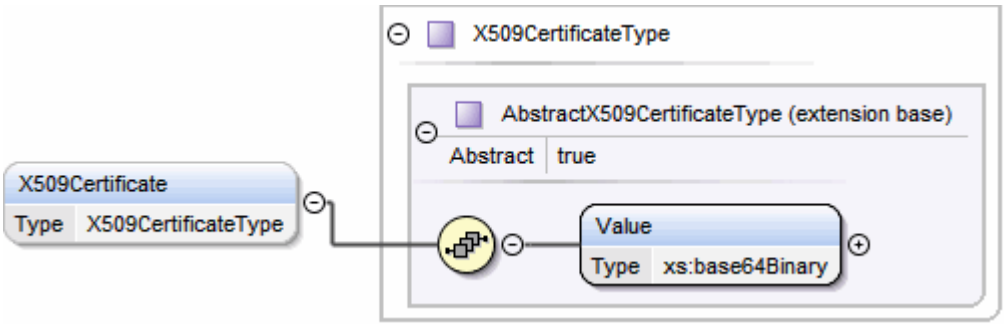


Beschreibung

Enthält den Schlüssel eines Zertifikates in Base64-Kodierung.

8.2 Element X509Certificate

	xplg:X509Certificate
Inhalt	Beschreibt ein X509-Zertifikat
Datentyp	`##elements`
Benutzt von	`xplg:Signature`

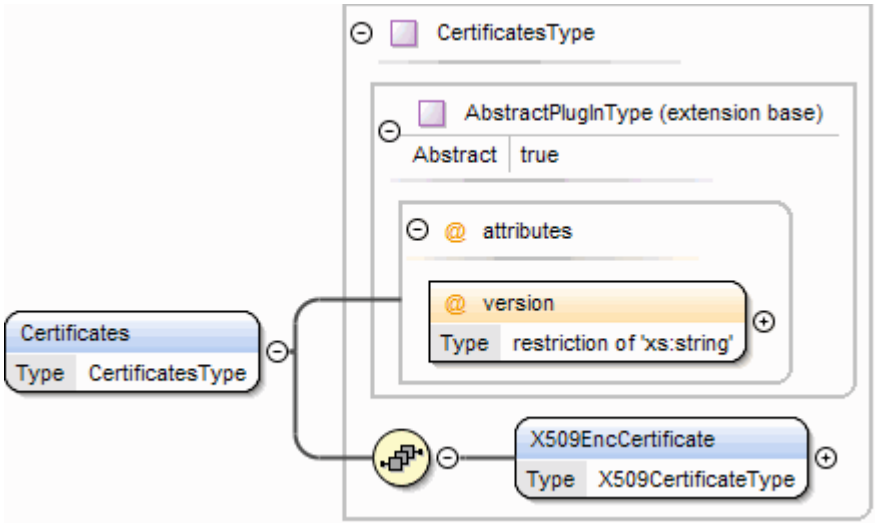


Beschreibung

Beschreibt ein X509-Zertifikat.

8.3 Plug-In Certificates

	`xplg:Certificates`
Inhalt	Beschreibt X509-Zertifikate, z.B. für die Verschlüsselung asynchroner Nachrichten an den ursprünglicher Absender.
Datentyp	`##elements`
Benutzt von	`xreq:MessagePlugIns`, `xreq:PackagePlugIns`, `xreq:TransportPlugIns`, `xres:MessagePlugIns`, `xres:PackagePlugIns`, `xres:TransportPlugIns`



Attribute

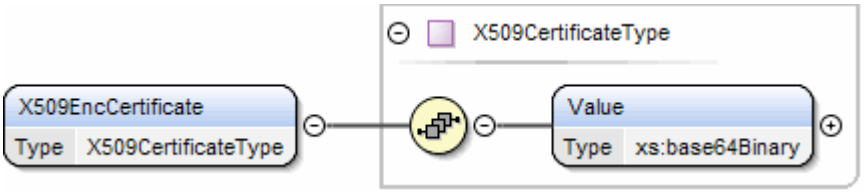
	`@version`	*optional*
Inhalt	Enthält die Versionsbezeichnung des Plug-Ins, derzeit 1.0.	
Datentyp	`xsd:string`	

Beschreibung

Beschreibt X509-Zertifikate, z.B. für die Verschlüsselung asynchroner Nachrichten an den ursprünglicher Absender.

8.4 Element X509EncCertificate

	xplg:X509EncCertificate
Inhalt	Beschreibt ein X509-Zertifikat für Verschlüsselungszwecke.
Datentyp	`##elements`
Benutzt von	`xplg:Certificates`

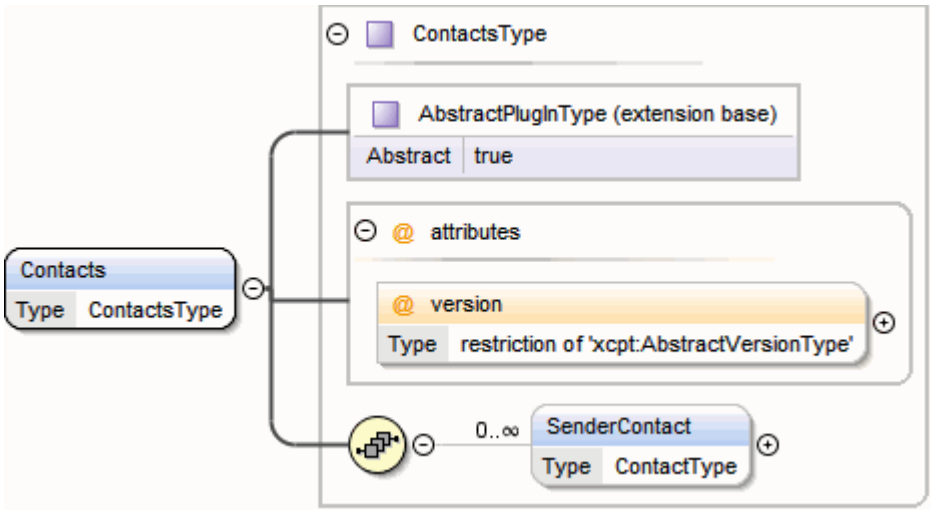


Beschreibung

Beschreibt ein X509-Zertifikat für Verschlüsselungszwecke.

8.5 Plug-In Contacts

	xplg:Contacts
Inhalt	Enthält eine beliebige Anzahl von Kontaktangaben
Datentyp	`##elements`
Benutzt von	`xreq:MessagePlugIns`, `xreq:PackagePlugIns`, `xreq:TransportPlugIns`, `xres:MessagePlugIns`, `xres:PackagePlugIns`, `xres:TransportPlugIns`



Attribute

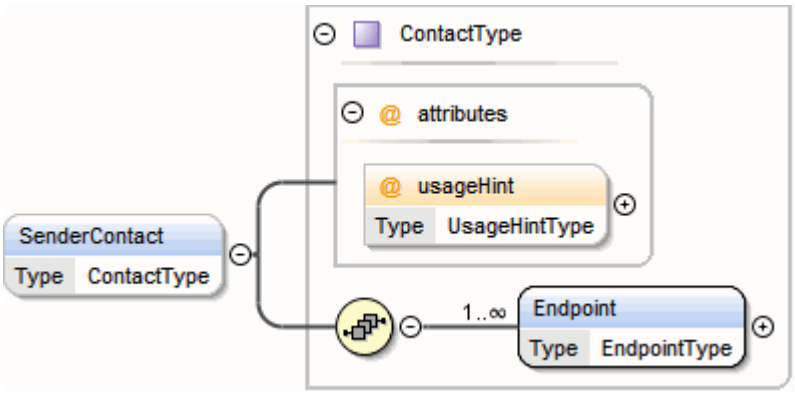
	@version	*erforderlich*
Inhalt	Enthält die Versionsbezeichnung des Plug-Ins, derzeit 1.0.	
Datentyp	`xsd:string`	

Beschreibung

Das Kindelement `SenderContact` definiert einen Kontakt im Bereich des Absenders. Die Anzahl von Kontakten ist beliebig.

8.6 Element SenderContact

	xplg:SenderContact
Inhalt	Beschreibt einen Kontakt im Bereich des Absenders
Datentyp	`##elements`
Benutzt von	`xplg:Communication`



Attribute

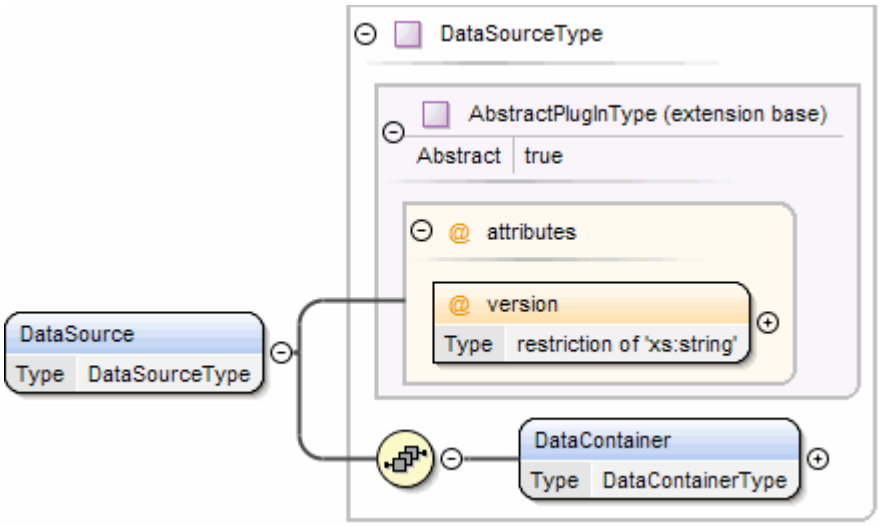
	@usageHint	*optional*
Inhalt	Ein Hinweis für die Verwendung des Kontaktes.	
Datentyp	`xsd:string`	

Beschreibung

Beschreibt einen Kontakt im Bereich des Absenders. Ein Kontakt kann mittels eines oder mehrerer Endpunkte (`Endpoint`) erreichbar sein. Das Attribut `usageHint` kann einen Hinweis liefern, in welchem Fall eine Kontaktaufnahme erfolgen soll.

8.7 Plug-In DataSource

	xplg:DataSource
Inhalt	Beschreibt in abstrakter Form die Quelle, aus der Nutzdaten stammen.
Datentyp	`##elements`
Benutzt von	`xreq:MessagePlugIns`, `xreq:PackagePlugIns`, `xreq:TransportPlugIns`, `xres:MessagePlugIns`, `xres:PackagePlugIns`, `xres:TransportPlugIns`



Attribute

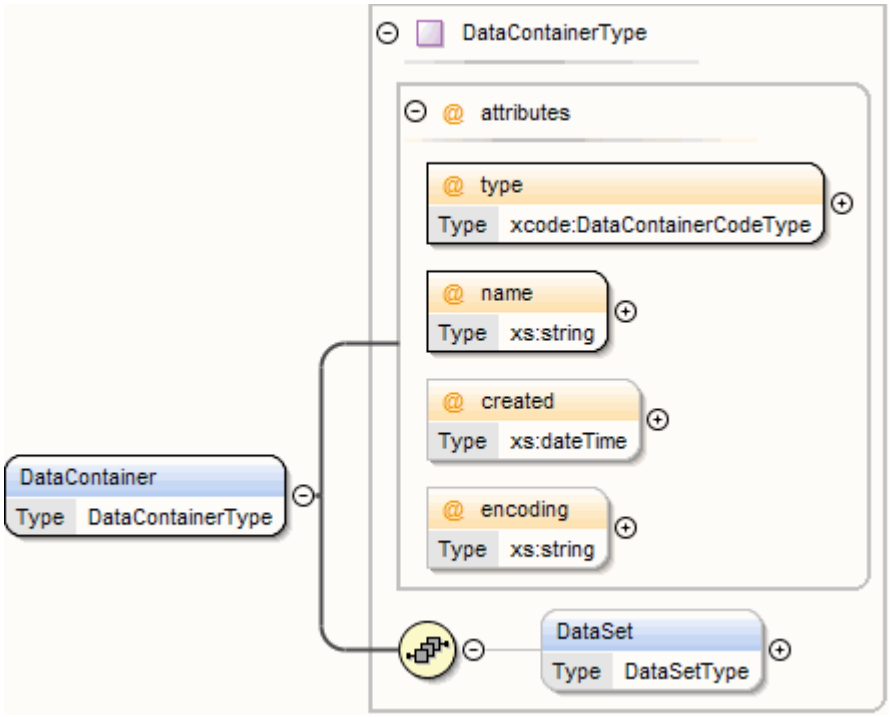
	@version	*optional*
Inhalt	Enthält die Versionsbezeichnung des Plug-Ins, derzeit 1.0.	
Datentyp	`xsd:string`	

Beschreibung

Beschreibt in abstrakter Form die Quelle, aus der Nutzdaten stammen. Es kann sich dabei z.B. um eine Datei oder eine Datenbank handeln.

8.8 Element DataContainer

	`xplg:DataContainer`
Inhalt	Beschreibt in abstrakter Form den Speicher, dem die Nutzdaten entnommen wurden.
Datentyp	`##elements`
Benutzt von	`xplg:DataSource`



Beschreibung

Beschreibt in abstrakter Form den Speicher, dem die Nutzdaten entnommen wurden. Dabei kann es sich um eine Datei oder eine Datenbank handeln. Das optionale Kindelement `Data-Set` ermöglicht die Beschreibung der Nutzdaten.

Attribute

	`@type`	*erforderlich*
Inhalt	Typ des Containers	
Datentyp	`Codeliste` *`DataContainerCode`*	

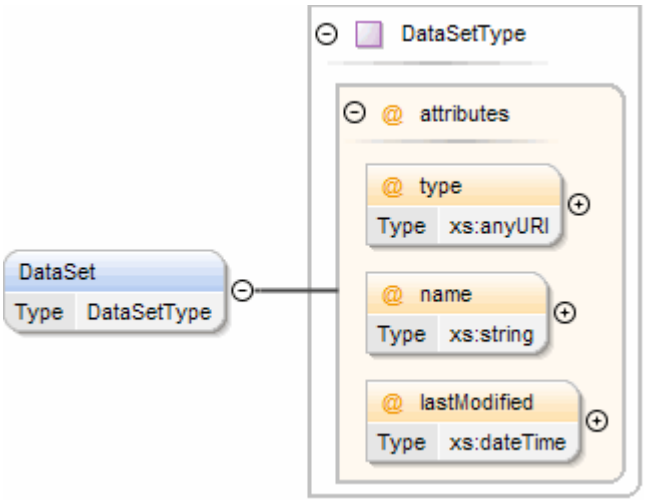
	`@name`	*erforderlich*
Inhalt	Bezeichnung des Containers	
Datentyp	`xsd:string`	

	`@created`	*optional*
Inhalt	Datum, an dem der Container erzeugt wurde.	

Datentyp	`xsd:dateTime`	
	`@encoding`	*optional*
Inhalt	Bezeichnung des Zeichensatzes	
Datentyp	`xsd:string`	

8.9 Element DataSet

	xplg:DataSet
Inhalt	Beschreibt Nutzdaten.
Datentyp	`##empty`
Benutzt von	`xplg:DataContainer`



Beschreibung

Beschreibt die Nutzdaten.

qAttribute

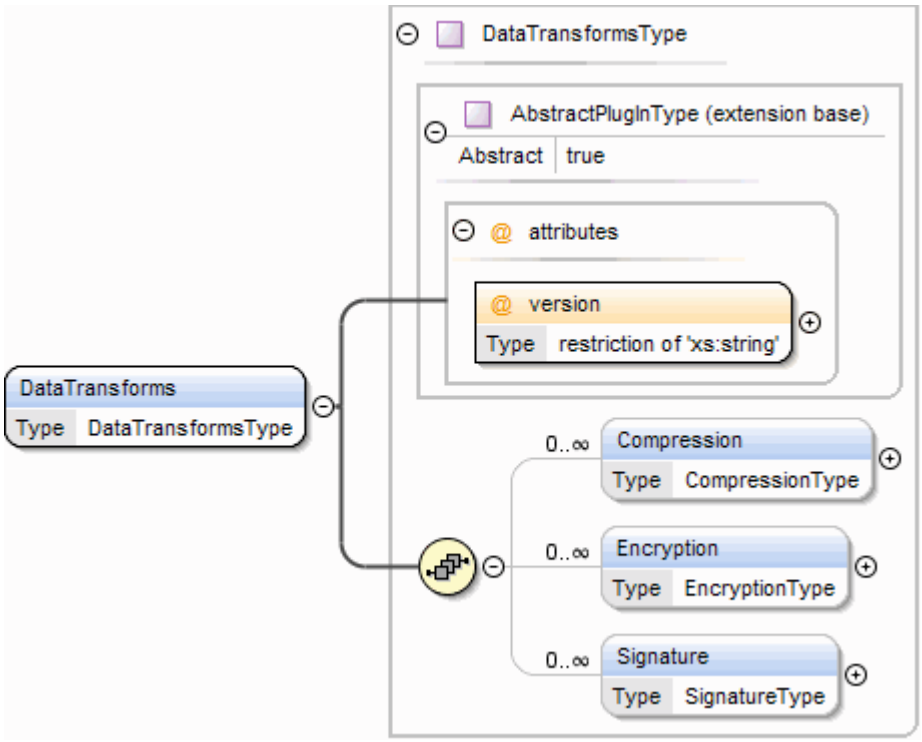
	@type	*erforderlich*
Inhalt	Typ der Nutzdaten	
Datentyp	`xsd:anyURI`	

	@name	*erforderlich*
Inhalt	Bezeichnung der Nutzdaten (z.B. Datensatzname)	
Datentyp	`xsd:string`	

	@lastModified	*optional*
Inhalt	Datum, an dem die Nutzdaten zuletzt aktualisiert wurden.	
Datentyp	`xsd:dateTime`	

8.10 Plug-In DataTransforms

	xplg:DataTransforms
Inhalt	Beschreibt Transformationen wie Verschlüsselung und Komprimierung.
Datentyp	`##elements`
Benutzt von	`xreq:MessagePlugIns`, `xreq:PackagePlugIns`, `xreq:TransportPlugIns`, `xres:MessagePlugIns`, `xres:PackagePlugIns`, `xres:TransportPlugIns`



Attribute

	@version	*optional*
Inhalt	Enthält die Versionsbezeichnung des Plug-Ins, derzeit 1.2.	
Datentyp	`xsd:string`	

Beschreibung

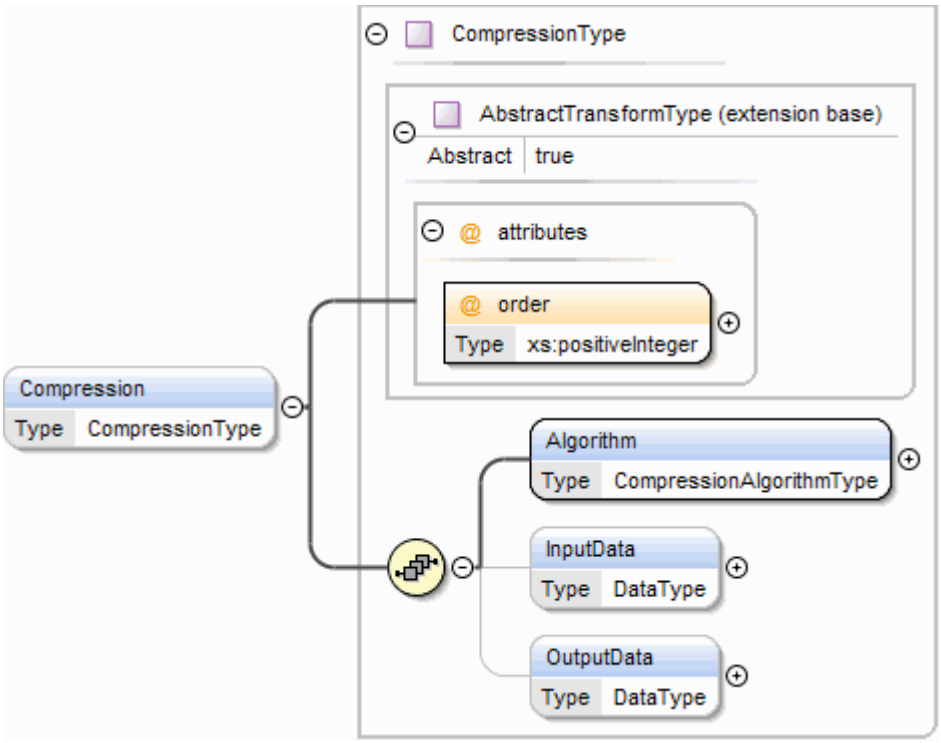
Beschreibt auf Nutzdaten oder andere Teile eines eXTra-Dokumentes bezogene Aktionen mittels spezifischer Kindelemente. Darstellbar sind die Transformationen Verschlüsselung und Komprimierung sowie die Signierung von Daten. Eine Aktion ist daher nicht zwingend modifizierend.

Jedes der Kindelement `Compression`, `Encryption` und `Signature` hat ein erforderliches Attribut `order`, dessen Wert eine positive Ganzzahl ist. In Abhängigkeit von der Verwendung

des Elementes TransformedData wird `order` entweder als Ordnungsnummer interpretiert, die die Reihenfolge der Aktionen angibt, oder als Identifikator einer Aktion.

8.11 Element Compression

	`xplg:Compression`
Inhalt	Beschreibt eine Aktion von Typ Komprimierung
Datentyp	`##elements`
Benutzt von	`xplg:DataTransforms`



Beschreibung

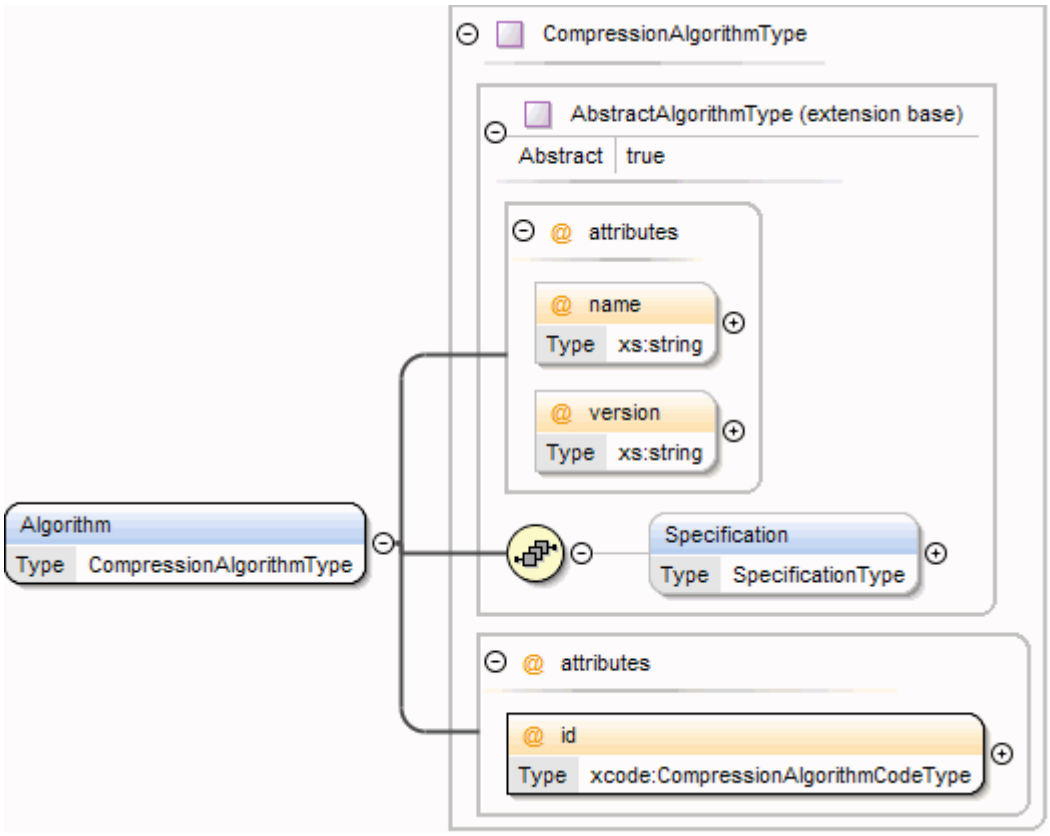
Beschreibt eine Aktion von Typ Komprimierung.

Attribute

	`@order`	*erforderlich*
Inhalt	Identifikator bzw. Ordnungsnummer der Aktion.	
Datentyp	`xsd:positiveInteger`	

8.12 Element Compression/Algorithm

	`xplg:Algorithm`
Inhalt	Bezeichnet einen Komprimierungsalgorithmus
Datentyp	`##elements`
Benutzt von	`xplg:Compression`



Beschreibung

Bezeichnet einen Komprimierungsalgorithmus.

Attribute

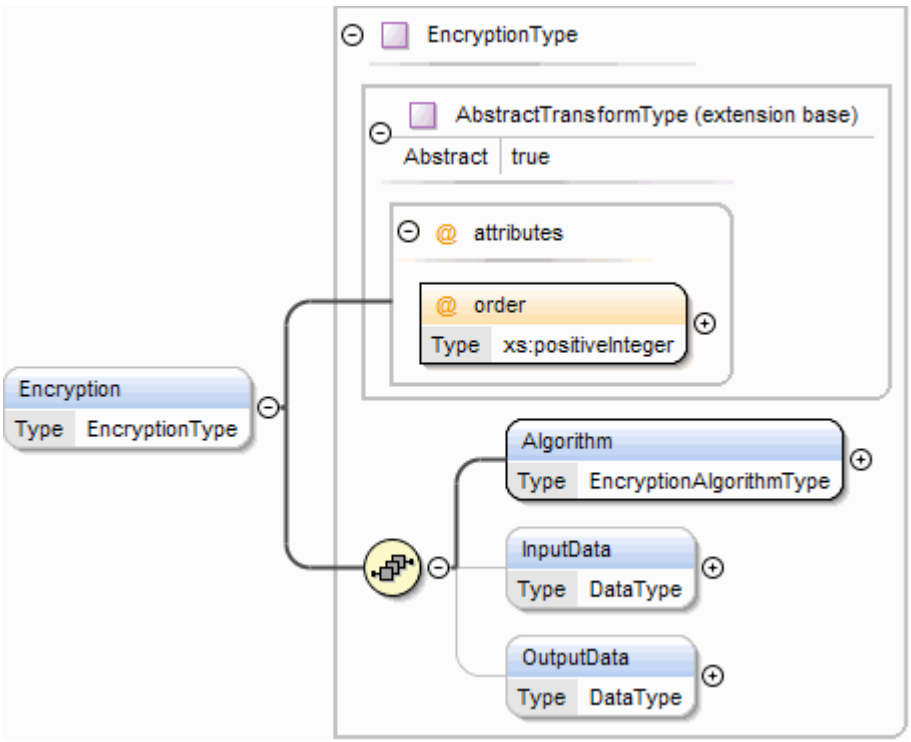
	`@id`	*Erforderlich*
Inhalt	Formaler Identifikator des Algorithmus	
Datentyp	`Codeliste` *`CompressionAlgorithmCode`*	

	`@name`	*Optional*
Inhalt	Bezeichner des Algorithmus.	
Datentyp	`xsd:string`	

	`@version`	*Optional*
Inhalt	Version des Algorithmus.	
Datentyp	`xsd:string`	

8.13 Element Encryption

	`xplg:Encryption`
Inhalt	Beschreibt eine Aktion von Typ Verschlüsselung
Datentyp	`##elements`
Benutzt von	`xplg:DataTransforms`



Beschreibung

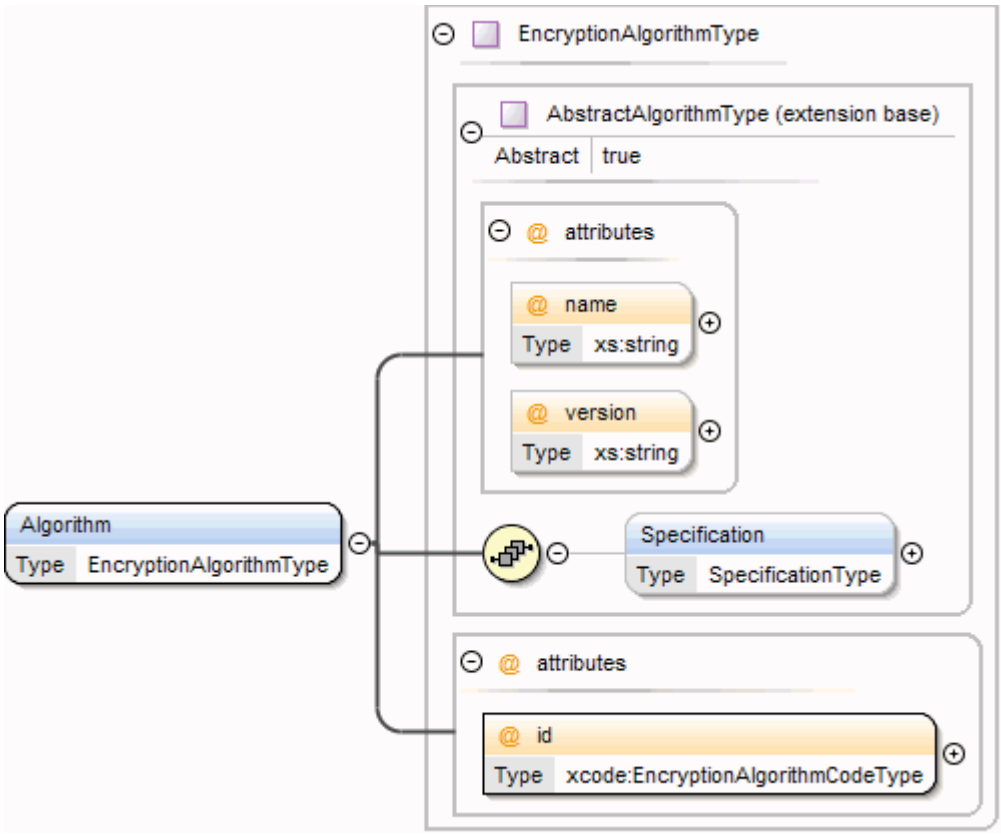
Beschreibt eine Aktion vom Typ Verschlüsselung.

Attribute

	`@order`	*Erforderlich*
Inhalt	Identifikator bzw. Ordnungsnummer der Aktion.	
Datentyp	`xsd:positiveInteger`	

8.14 Element Encryption/Algorithm

	`xplg:Algorithm`
Inhalt	Bezeichnet einen Verschlüsselungsalgorithmus
Datentyp	`##elements`
Benutzt von	`xplg:Encryption`



Beschreibung

Bezeichnet einen Verschlüsselungsalgorithmus.

Attribute

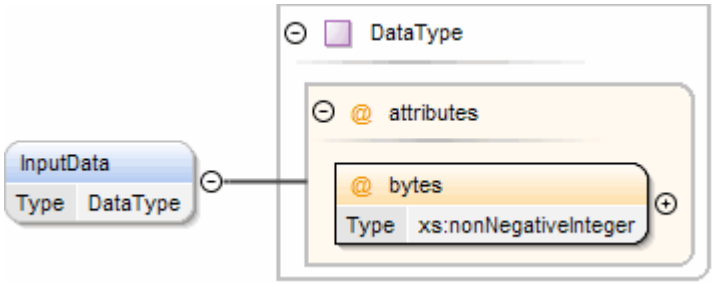
	`@id`	*Erforderlich*
Inhalt	Formaler Identifikator des Algorithmus	
Datentyp	`Codeliste` *`EncryptionAlgorithmCode`*	

	`@name`	*Optional*
Inhalt	Bezeichner des Algorithmus.	
Datentyp	`xsd:string`	

	`@version`	*Optional*
Inhalt	Version des Algorithmus.	
Datentyp	`xsd:string`	

8.15 Element InputData

	`xplg:InputData`
Inhalt	Beschreibt den Zustand von Daten *vor* einer Transformation.
Datentyp	`##elements`
Benutzt von	`xplg:Compression`, `xplg:Encryption`



Beschreibung

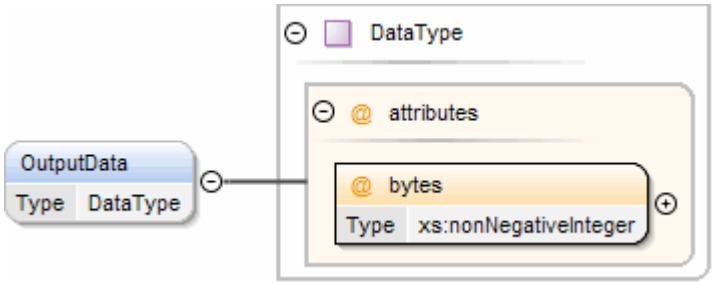
Beschreibt den Zustand von Daten vor einer Transformation. Zur Zeit kann die Größe der Daten in Bytes angegeben werden.

Attribute

	`@bytes`	*Erforderlich*
Inhalt	Größe der Daten in Bytes	
Datentyp	`xsd:nonNegativeInteger`	

8.16 Element OutputData

	`xplg:OutputData`
Inhalt	Beschreibt den Zustand von Daten *nach* einer Transformation.
Datentyp	`##elements`
Benutzt von	`xplg:Compression`, `xplg:Encryption`



Beschreibung

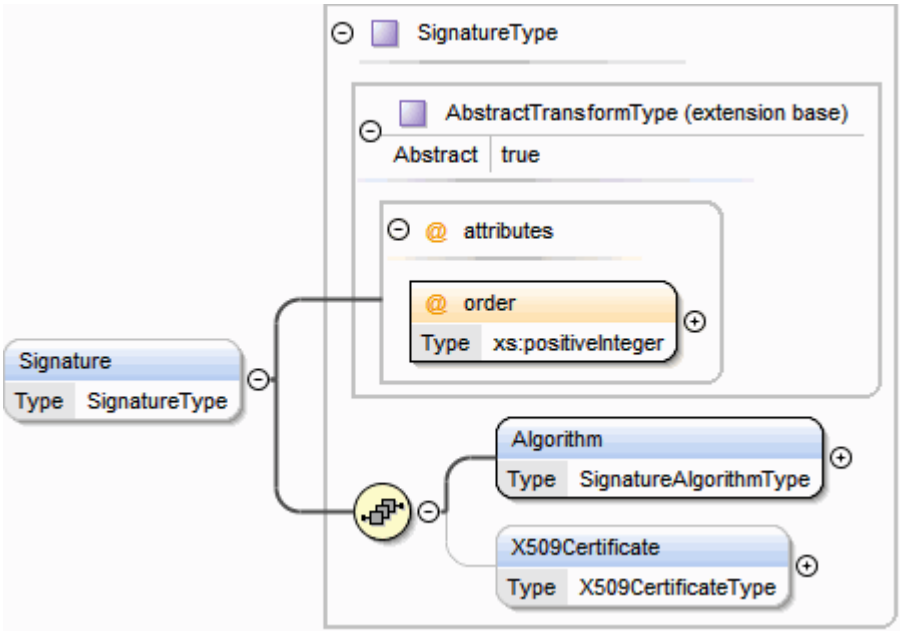
Beschreibt den Zustand von Daten nach einer Transformation. Zur Zeit kann die Größe der Daten in Bytes angegeben werden.

Attribute

	`@bytes`	*Erforderlich*
Inhalt	Größe der Daten in Bytes	
Datentyp	`xsd:nonNegativeInteger`	

8.17 Element Signature

	xplg:Signature
Inhalt	Beschreibt eine (an den Nutzdaten erfolgte) Signatur
Datentyp	`##elements`
Benutzt von	`xplg:DataTransforms`



Beschreibung

Beschreibt eine Aktion von Typ Signierung. Das verwendete Zertifikat kann optional beigefügt werden.

Attribute

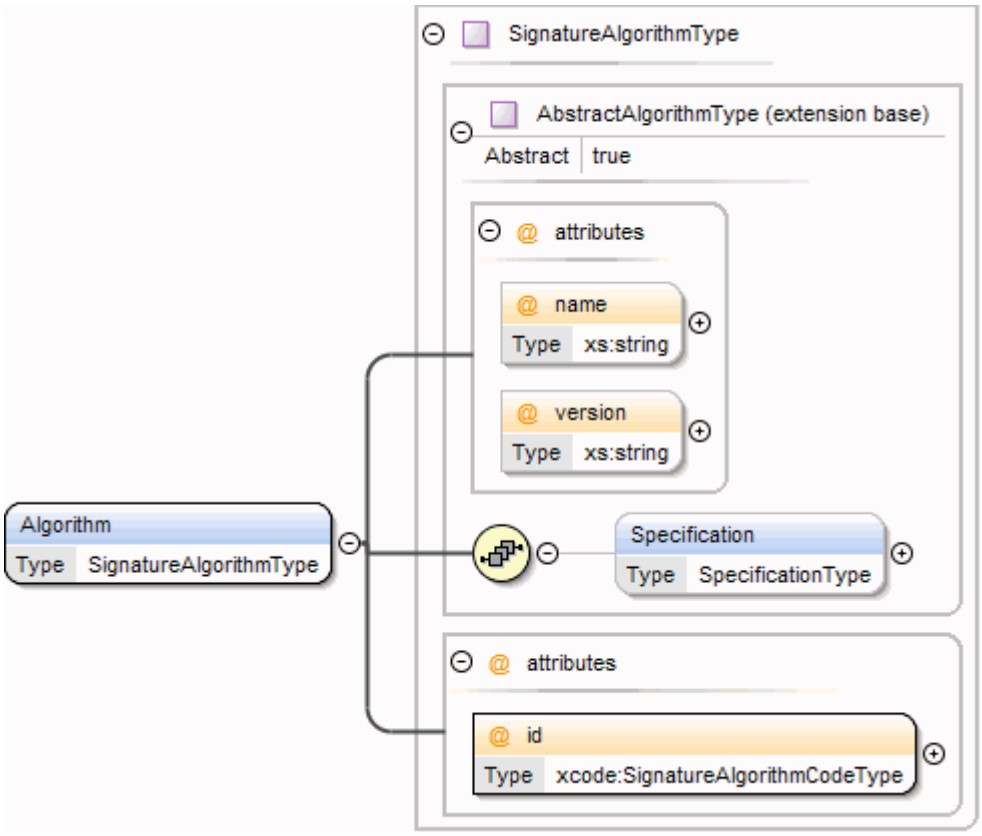
	@order	*Erforderlich*
Inhalt	Identifikator bzw. Ordnungsnummer der Aktion.	
Datentyp	`xsd:positiveInteger`	

	@url	*Optional*
Inhalt	URL der Spezifikation	
Datentyp	`xsd:anyURI`	

	@version	*Optional*
Inhalt	Version der Spezifikation	
Datentyp	`xsd:string`	

8.18 Element Signature/Algorithm

	xplg:Algorithm
Inhalt	Bezeichnet ein Signaturverfahren.
Datentyp	`##elements`
Benutzt von	`xplg:Signature`



Beschreibung

Bezeichnet ein Signaturverfahren.

Attribute

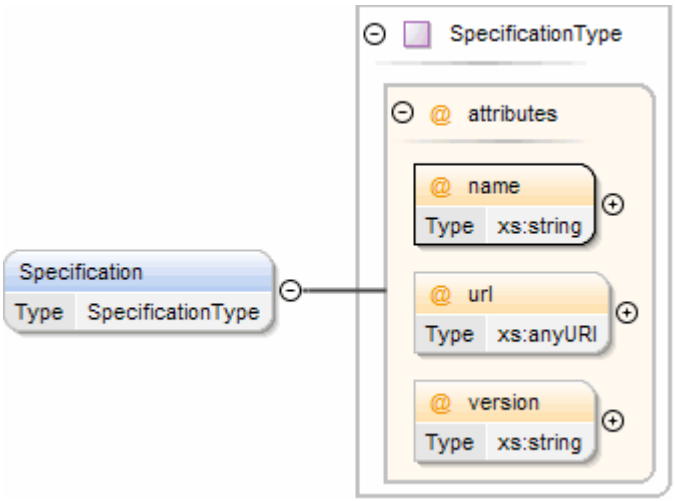
	@id	*Erforderlich*
Inhalt	Formaler Identifikator des Verfahrens.	
Datentyp	`Codeliste` *SignatureAlgorithmCode*	

	@name	*Optional*
Inhalt	Bezeichner des Verfahrens.	
Datentyp	`xsd:string`	

	@version	*Optional*
Inhalt	Version des Verfahrens.	
Datentyp	`xsd:string`	

8.19 Element Specification

	xplg:Specification
Inhalt	Referenziert eine Spezifikation oder andere Beschreibung eines Algorithmus
Datentyp	`##empty`
Benutzt von	`xplg:Algorithm`



Beschreibung

Ermöglicht es, für einen Algorithmus zu Dokumentationszwecken eine Referenz auf eine Spezifikation oder andere Beschreibung abzulegen.

Attribute

	@name	*erforderlich*
Inhalt	Bezeichnung der Spezifikation	
Datentyp	`xsd:string`	

	@url	*optional*
Inhalt	URL der Spezifikation	
Datentyp	`xsd:anyURI`	

	@version	*optional*
Inhalt	Version der Spezifikation	
Datentyp	`xsd:string`	

9 Logging [normativ]

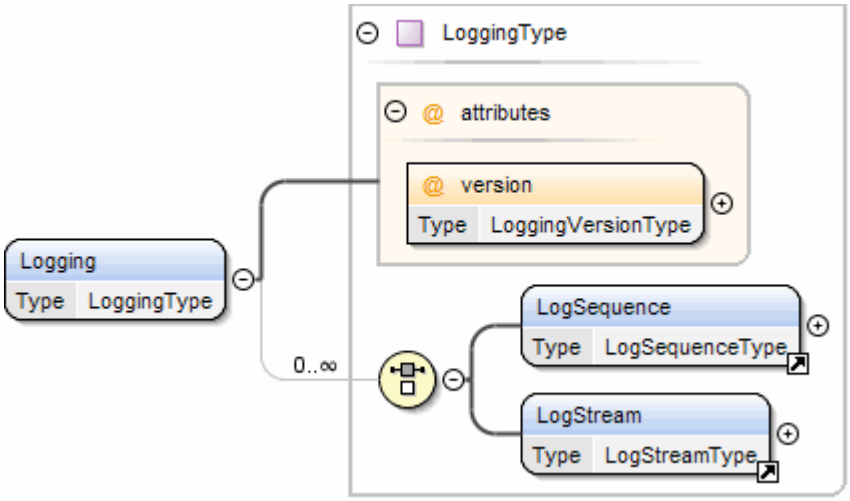
Logging ist eine Komponente für die Protokollierung des Transports und der Verarbeitung von eXTra-Dokumenten, und zur Unterstützung der Dokumentation des laufenden Betriebes und der Fehleranalyse. Die Namensraum-URI ist

http://www.extra-standard.de/namespace/logging/1

Der lokale Name des Wurzelelementes ist `Logging`. Es wird empfohlen, den Namensraum-präfix `xlog` zu verwenden.

9.1 Element Logging

	`xlog:Logging`
Inhalt	Enthält das Log einer logischen Ebene des eXTra-Modells
Datentyp	`##elements`
Benutzt von	`xreq:Message`, `xreq:Package`, `xreq:XMLTransport`, `xres:Message`, `xres:Package`, `xres:XMLTransport`



Attribute

	`@version` *optional*
Inhalt	Enthält die Versionsbezeichnung des eXTra-Schemas. *Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden*.
Datentyp	`xsd:anyURI`

Profilierung

Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`xlog:LogSequence`	+	+	+		+	
`xlog:LogStream`	+	+	+		+	

Beschreibung

`xlog:Logging` ist das Wurzelelement eines Logs. Jede Ebene des eXTra-Modells wird separat geloggt.

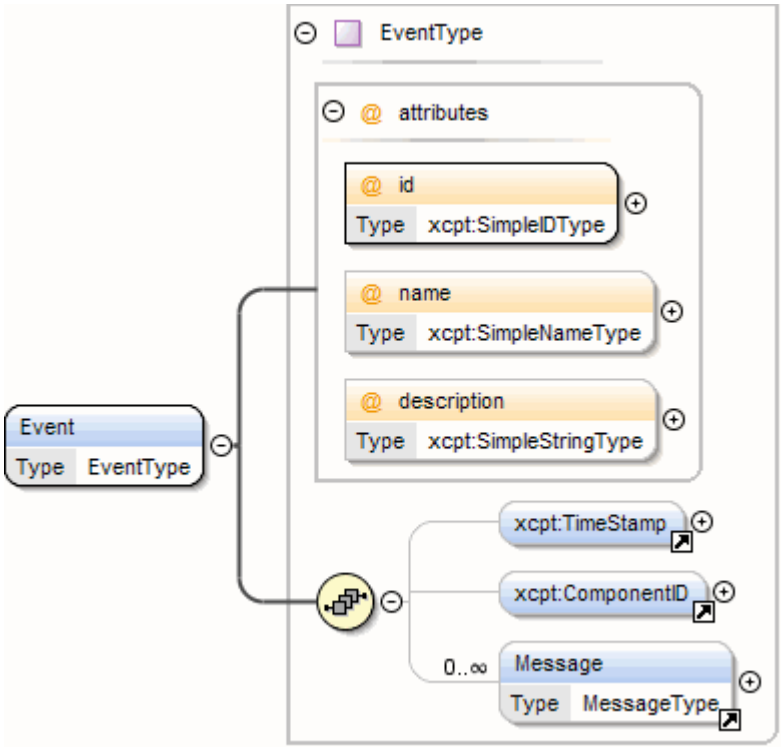
Ein Log enthält die Elemente `xlog:LogSequence` und `xlog:LogStream` in beliebiger Folge und Häufigkeit.

Beispiel

```
<xlog:Logging version="1.0">
    <xlog:LogSequence>
        <xcpt:TimeStamp>2013-06-06T00:00:01</xcpt:TimeStamp>
        <xcpt:ComponentID>CORE.KONVERT 1.5</xcpt:ComponentID>
        <xlog:Event id="http://www.extra-standard.de/event/RECEIVE" name="Dateneingang">
            <xcpt:TimeStamp>2013-06-06T00:00:01</xcpt:TimeStamp>
            <xcpt:ComponentID>CORE.SERVER</xcpt:ComponentID>
        </xlog:Event>
        <xlog:State>
            <xcpt:TimeStamp>2013-06-06T00:00:01</xcpt:TimeStamp>
            <xcpt:Property name="generated.documentid"
                  type="xs:string">12968438573722894</xcpt:Property>
        </xlog:State>
        <xlog:Operation completionCode="0" successful="true"
           id="http://www.extra-standard.de/operation/VALIDATE_XML"
           description="Validierung Transferobjekt">
            <xlog:Parameter class="http://www.extra.standard.de/class/SCHEMA"
               usage="http://www.extra-standard.de/usage/IN"
               description="XSD Schema" >file://...</xlog:Parameter>
        </xlog:Operation>
        <xlog:Operation completionCode="0" successful="true"
           id="http://www.extra-standard.de/operation/DECRYPT"
           description="Entschluesselung">
            <xlog:Object class="http://www.extra.standard.de/class/XML_CONTENT"
               location="XMLTransport/TransferBody"/>
        </xlog:Operation>
        <xlog:Operation completionCode="12" successful="false"
            id="http://www.destatis.de/operation/VALIDATE"
            description="Pruefung der Rohdaten">
            <xcpt:ComponentID>CORE.INSPECTOR 1.5</xcpt:ComponentID>
            <xlog:Exception>
                <xcpt:TimeStamp>2013-06-06T00:00:01</xcpt:TimeStamp>
                <xlog:Message>
                    <xcpt:TimeStamp>2013-06-06T00:00:01</xcpt:TimeStamp>
                    <xcpt:Text>Invalid data</xcpt:Text>
                </xlog:Message>
            </xlog:Exception>
        </xlog:Operation>
    </xlog:LogSequence>
</xlog:Logging>
```

9.2 Element Event

	`xlog:Event`
Inhalt	Beschreibt ein Ereignis.
Datentyp	`##elements`
Benutzt von	`xlog:LogSequence`, `xlog:Operation`

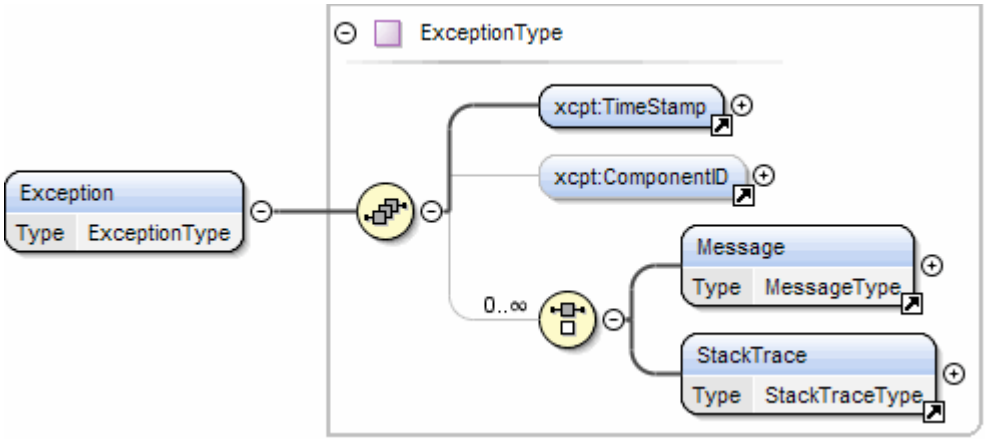


Beschreibung

`xlog:Event` beschreibt ein während der Verarbeitung einer Nachricht eingetretenes Ereignis. Das Ereignis ist über eine ID zu referenzieren. Die Klassifikation von Ereignissen und die Vergabe von IDs ist anwendungsspezifisch. Ein Ereignis kann mit einem Zeitstempel und einem Bezug zu einer Komponente (des Verarbeitungssystems) sowie einer mit der Ausnahme in Verbindung stehenden Nachricht genauer beschrieben werden.

9.3 Element Exception

	`xlog:Exception`
Inhalt	Beschreibt eine Ausnahme.
Datentyp	`##elements`
Benutzt von	`xlog:LogSequence`, `xlog:Operation`

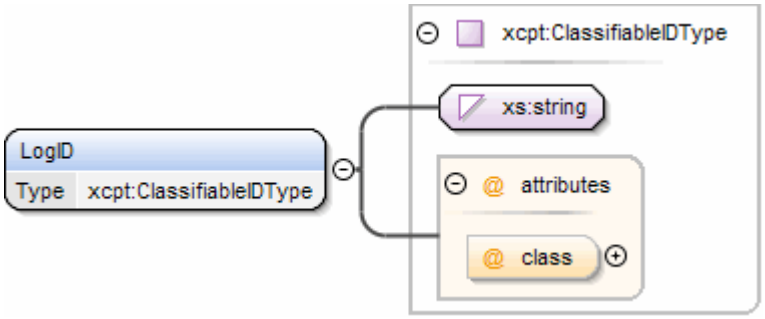


Beschreibung

`xlog:Exception` beschreibt eine bei der Verarbeitung einer Nachricht eingetretene Ausnahme, also eine Fehlersituation. Die Ausnahme muss mittels einer ID referenzierbar gemacht werden. Die Klassifikation von Ausnahmen und die Vergabe von IDs ist anwendungsspezifisch. Eine Ausnahme kann mit einem Zeitstempel und einem Bezug zu einer Komponente (des Verarbeitungssystems) sowie einer mit der Ausnahme in Verbindung stehenden Nachrichten genauer beschrieben werden.

9.4 Element LogID

	`xlog:LogID`
Inhalt	Enthält die ID des aktuellen `xlog:LogSequence` bzw. `xlog:LogStream`
Datentyp	`xsd:string`
Benutzt von	`xlog:LogTrace`



Attribute

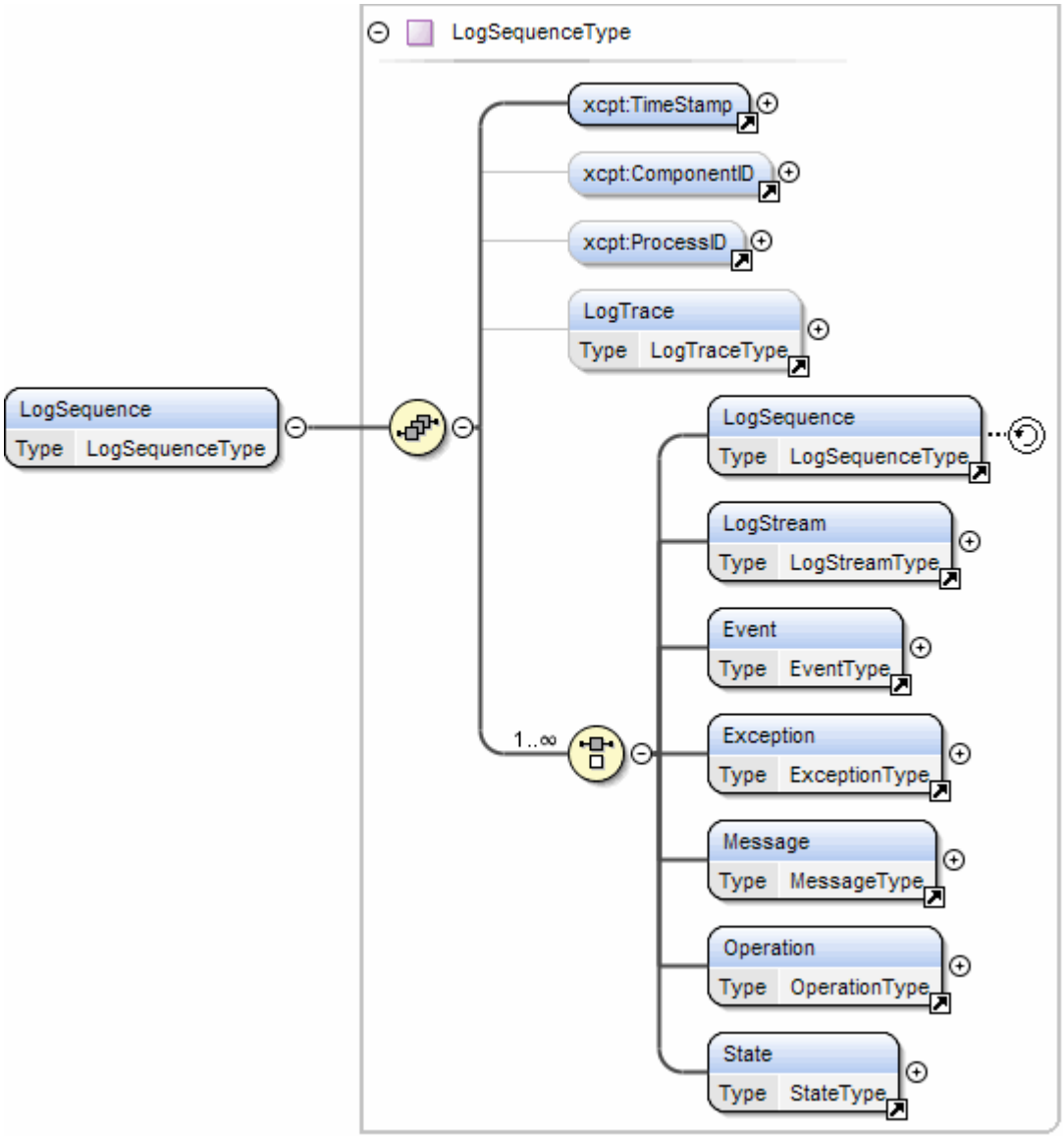
	`@class`	*optional*
Inhalt	Eine Angabe zur Klassifizierung der ID.	
Datentyp	`xsd:string`	

Beschreibung

Mit `xlog:LogID` kann einem `xlog:LogSequence` bzw. `xlog:LogStream` Element eine ID zugeordnet werden.

9.5 Element LogSequence

	`xlog:LogSequence`
Inhalt	Beschreibt eine [zusammenhängende] Folge von log-fähigen Objekten und Ereignissen mittels dezidierter Elemente.
Datentyp	`##elements`
Benutzt von	`xlog:Logging`, `xlog:LogSequence`

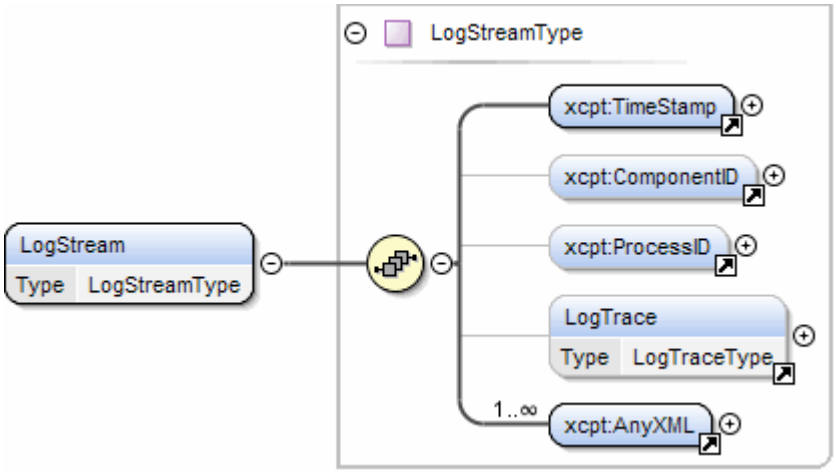


Beschreibung

`xlog:LogSequence` dient der Beschreibung einer zusammenhängenden Folge logrelevanter Ereignisse und Objekte.

9.6 Element LogStream

	`xlog:LogStream`
Inhalt	Enthält beliebig formatierte Logeinträge
Datentyp	`##elements`
Benutzt von	`xlog:Logging`, `xlog:LogSequence`

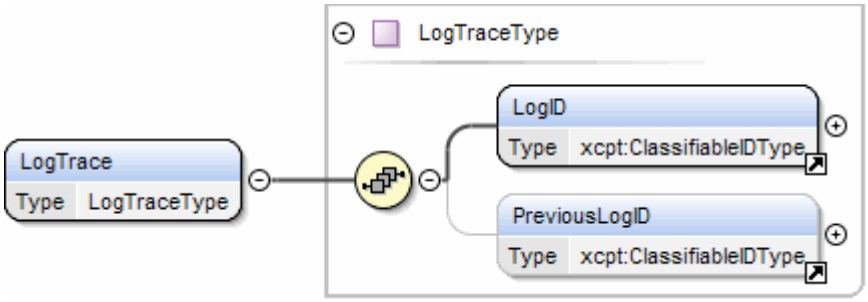


Beschreibung

Mit `xlog:LogStream` haben Anwender die Möglichkeit, vorhandene Log-Formate ohne Umstellungszwang weiter zu verwenden. Solche Logs sind in der Regel anwendungsspezifisch. Im Gegensatz zu `xlog:LogSequence` ist es mit `xlog:LogStream` deshalb nicht möglich, Logs mit einheitlicher Software auszuwerten.

9.7 Element LogTrace

	xlog:LogTrace
Inhalt	Verknüpft Instanzen von `xlog:LogSequence` und `xlog:LogStream`.
Datentyp	`##elements`
Benutzt von	`xlog:LogSequence`, `xlog:LogStream`

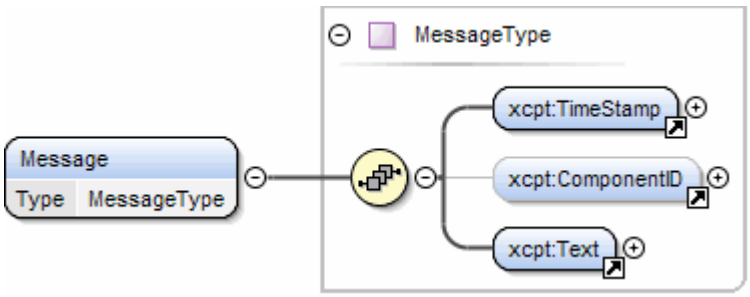


Beschreibung

`xlog:LogTrace` dient der Verknüpfung von Instanzen von `xlog:LogSequence` und `xlog:LogStream`, und hierbei insbesondere der Rückverfolgung durch Zugriff auf historisch vorausgehende Logeinträge.

9.8 Element Message

	xlog:Message
Inhalt	Beschreibt eine einem Log-Objekt zugeordnete Nachricht.
Datentyp	`##elements`
Benutzt von	`xlog:Event`, `xlog:Exception`, `xlog:LogSequence`, `xlog:Operation`

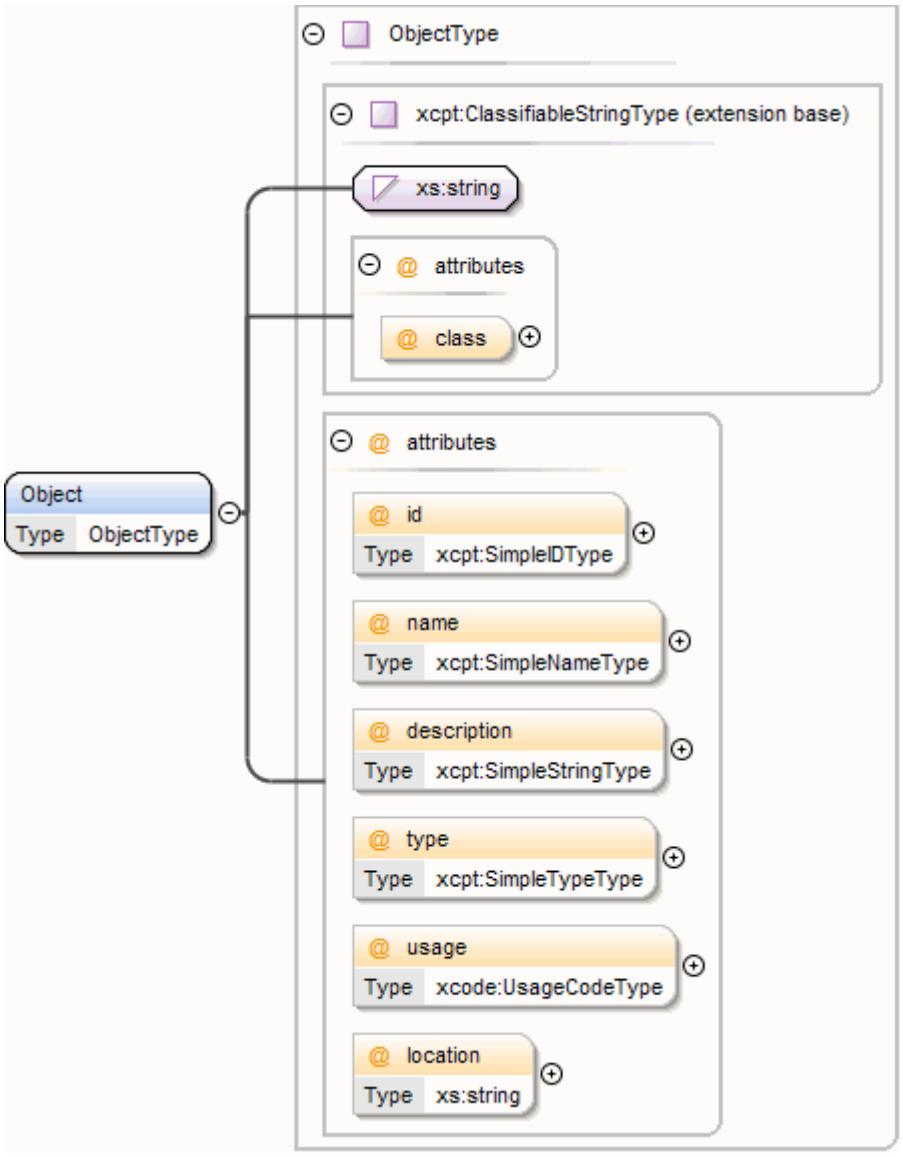


Beschreibung

`xlog:Message` enthält eine Nachricht im Kontext eines Ereignisses, einer Ausnahme, einer Operation oder einer Log-Sequenz.

9.9 Element Object

	`xlog:Object`
Inhalt	Beschreibt ein Objekt im Kontext einer Operation..
Datentyp	`##empty`
Benutzt von	`xlog:Operation`



Attribute

	`@class`	*optional*
Inhalt	Eine Angabe zur Klassifizierung des Objekts.	
Datentyp	`xsd:string`	

	`@id`	*optional*
Inhalt	Eine Angabe zur Identifizierung des Objekts.	
Datentyp	`xsd:token`	

	@name	*optional*
Inhalt	Der Name des Objekts.	
Datentyp	`xsd:string`	

	@description	*optional*
Inhalt	Eine (kurze) Beschreibung Objekts.	
Datentyp	`xsd:string`	

	@type	*optional*
Inhalt	Eine Angabe zur Typisierung des Objekts.	
Datentyp	`xsd:string`	

	@usage	*optional*
Inhalt	Eine Angabe zur Art der Verwendung des Objekts.	
Datentyp	Codeliste *UsageCode*	

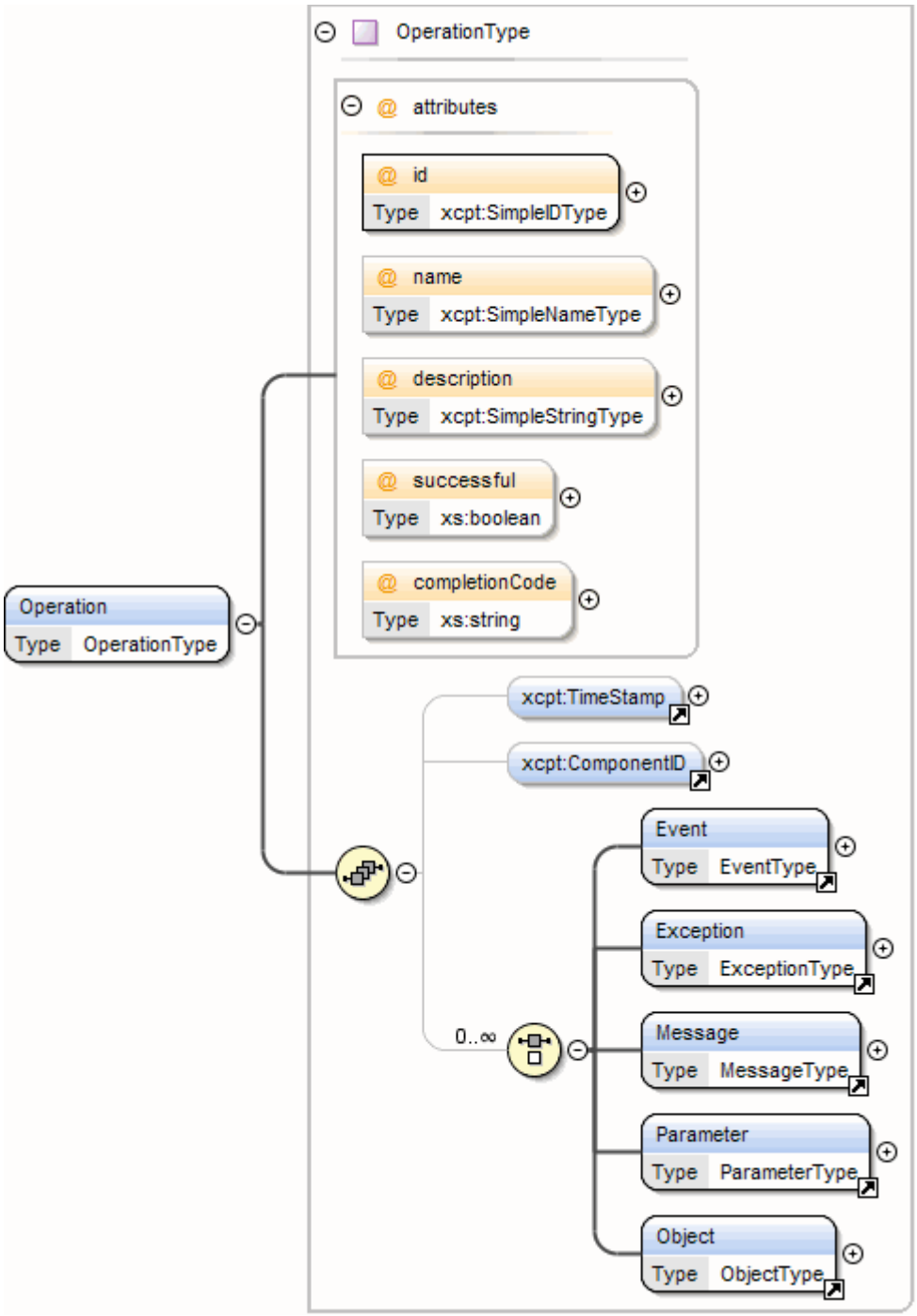
	@location	*optional*
Inhalt	Eine Angabe zur Lokaliserung des Objekts.	
Datentyp	`xsd:string`	

Beschreibung

Im Kontext einer Operation (`xlog:Operation`) können mittels `xlog:Object` beliebige involvierte Objekte beschrieben werden. Das Verständnis von Objekt ist dabei allgemein und nicht auf technische Konzepte der Objektorientierung beschränkt. Objekte können z.B. Dateien sein, oder Datenbanken, oder andere Ressourcen.

9.10 Element Operation

	`xlog:Operation`
Inhalt	Beschreibt eine Operation.
Datentyp	`##elements`
Benutzt von	`xlog:LogSequence`



Attribute

	@id	*required*
Inhalt	Eine Angabe zur Identifizierung des Objekts.	
Datentyp	`xsd:token`	

	@name	*optional*
Inhalt	Der Name des Objekts.	
Datentyp	`xsd:string`	

	@description	*optional*
Inhalt	Eine (kurze) Beschreibung Objekts.	
Datentyp	`xsd:string`	

	@succesful	*optional*
Inhalt	Zeigt an, ob die Operation erfolgreich war.	
Datentyp	`xsd:boolean`	

	@completionCode	*optional*
Inhalt	Enthält einen Ausführungscode.	
Datentyp	`xsd:string`	

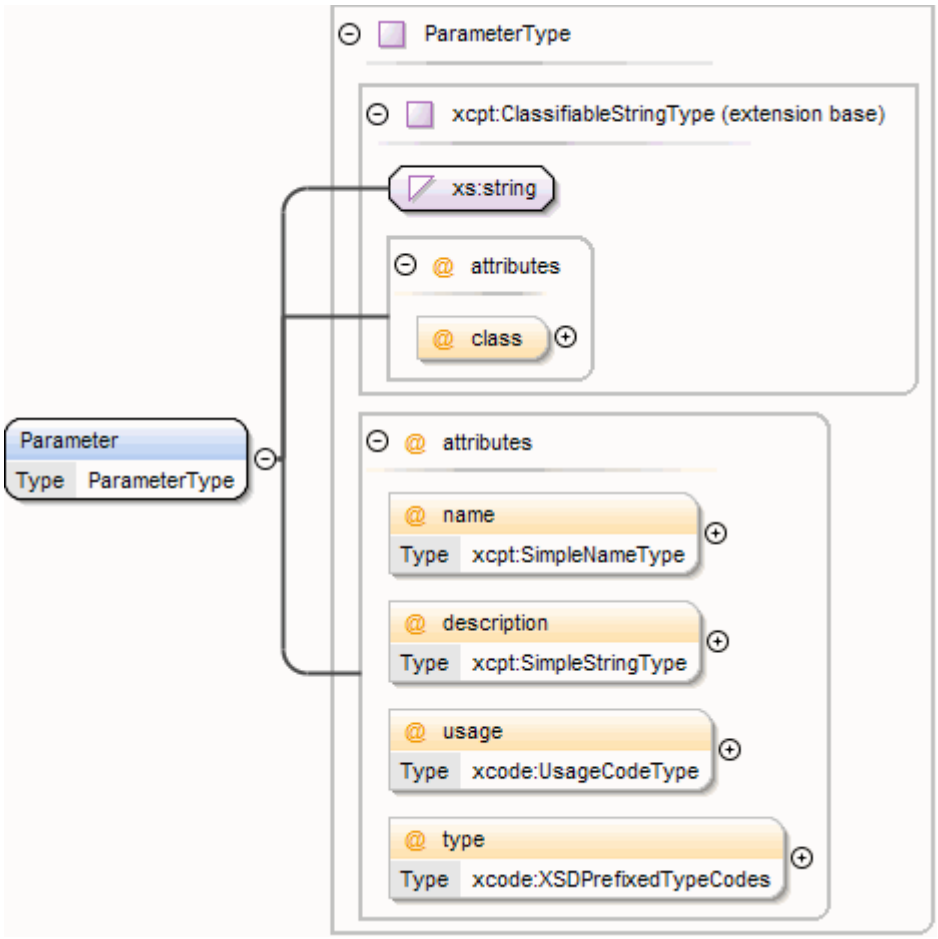
Beschreibung

`xlog:Operation` beschreibt eine Operation. Unter Operation ist jeder beliebige Verarbeitungsschritt zu verstehen. Es obliegt den Anwendern, Operationen mit einer geeigneten Granularität zu definieren.

Operationen müssen mit einer ID gekennzeichnet werden. Name, Beschreibung, Erfolgsanzeige und Ausführungscodes, sowie alle weiteren Angaben sind optional. Operationen können mit einem Zeitstempel versehen und einer Komponente zugeordnet werden. Desweiteren ist es möglich, mit der Operation verbundene Ereignisse, Ausnahmen, Nachrichten, Parameter und Objekte zu beschreiben.

9.11 Element Parameter

	`xlog:Parameter`
Inhalt	Beschreibt einen Parameter einer Operation.
Datentyp	`##empty`
Benutzt von	`xlog:Operation`



Attribute

	`@class`	*optional*
Inhalt	Eine Angabe zur Klassifizierung des Parameters.	
Datentyp	`xsd:string`	

	`@name`	*optional*
Inhalt	Der Name des Parameters.	
Datentyp	`xsd:string`	

	`@description`	*optional*
Inhalt	Eine (kurze) Beschreibung Parameters.	
Datentyp	`xsd:string`	

	`@type`	*optional*
Inhalt	Eine Angabe zur Datentyp des Parameters.	
Datentyp	`xsd:string`	

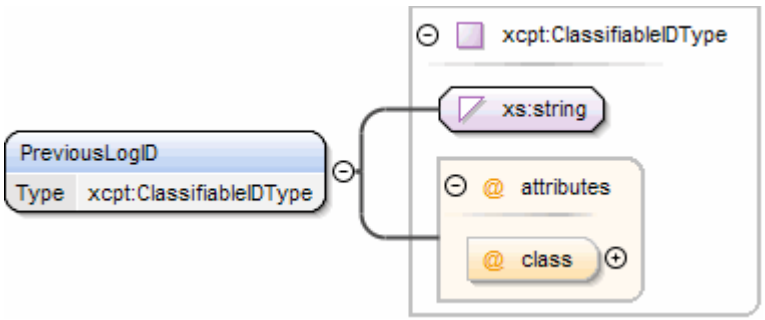
	`@usage`	*optional*
Inhalt	Eine Angabe zur Art der Verwendung des Objekts.	
Datentyp	Codeliste *UsageCode*	

Beschreibung

`xlog:Parameter` beschreibt einen Parameter einer Operation.

9.12 Element PreviousLogID

	xlog:PreviousLogID
Inhalt	Enthält die ID des historisch vorausgehenden `xlog:LogSequence` bzw. `xlog:LogStream`
Datentyp	`xsd:string`
Benutzt von	`xlog:LogTrace`



Attribute

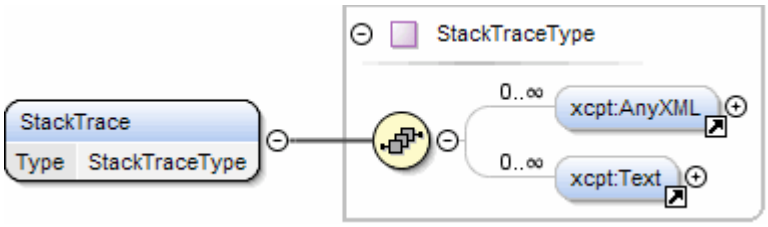
	@class	*optional*
Inhalt	Eine Angabe zur Klassifizierung der ID.	
Datentyp	`xsd:string`	

Beschreibung

Mit `xlog:LogID` kann einem `xlog:LogSequence` bzw. `xlog:LogStream` Element eine ID zugeordnet werden.

9.13 Element StackTrace

	xlog:StackTrace
Inhalt	Enthält die Repräsentation eines Laufzeitstacks.
Datentyp	`##elements`
Benutzt von	`xlog:Exception`

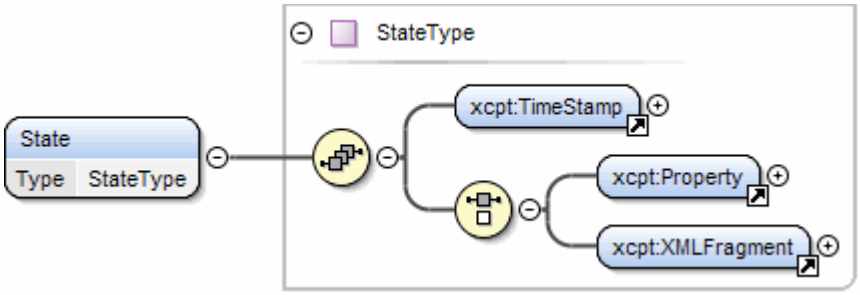


Beschreibung

`xlog:StackTrace` enthält die Repräsentation eines Laufzeitstacks in beliebiger XML-konformer Form.

9.14 Element State

	xlog:State
Inhalt	Beschreibt einen Zustand als Folge von Name/Wert-Paaren und/oder mit Hilfe von XML-Fragmenten
Datentyp	`##elements`
Benutzt von	`xlog:LogSequence`



Beschreibung

`xlog:State` beschreibt einen Zustand als Folge von Name/Wert-Paaren und/oder mit Hilfe von XML-Fragmenten. Zustände beziehen sich auf eine Logsequenz und können z.B. beliebige relevante Anfangs- und Endzustände beschreiben. Namen und Werte sind anwendungsspezifisch.

Beispiel

```
<xlog:State>
    <xlog:Property name="connectionAlive" type="xs:boolean">true</xlog:Property>
    <xlog:Property name="numberOutstandingRequest" type="xs:integer">12</xlog:Property>
</xlog:State>
```

10 Allgemeine Komponenten [normativ]

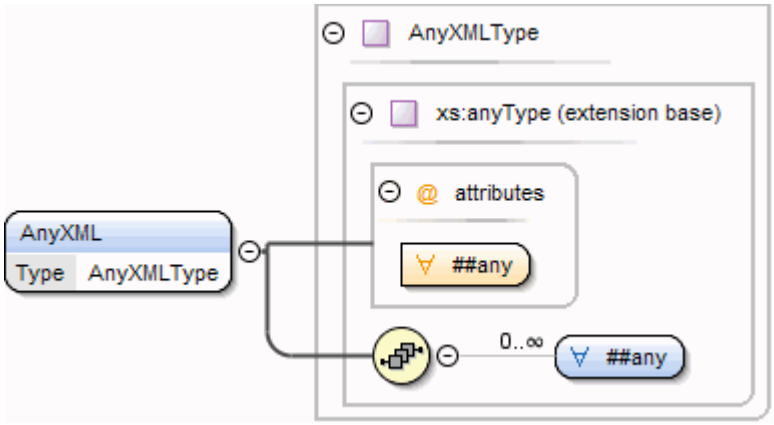
Unter den allgemeinen Komponenten sind solche Elemente zusammengefasst, die entweder per Definition von allgemeiner Bedeutung und nicht an eine spezifische ‚höhere' Komponente gebunden sind, oder die in mehr als einer anderen Komponente benutzt werden. Die Namensraum-URI ist

http://www.extra-standard.de/namespace/components/1

Es wird empfohlen, den Namensraumpräfix `xcpt` zu verwenden.

10.1 Element AnyXML

	`xcpt:AnyXML`
Inhalt	Beliebiger XML-konformer Inhalt
Datentyp	`xsd:anyType`
Benutzt von	`xcpt:DataType`, `xlog:LogStream`, `xlog:StackTrace`



Attribute

Aufgrund des Datentyps sind beliebige Attribute zulässig.

Beschreibung

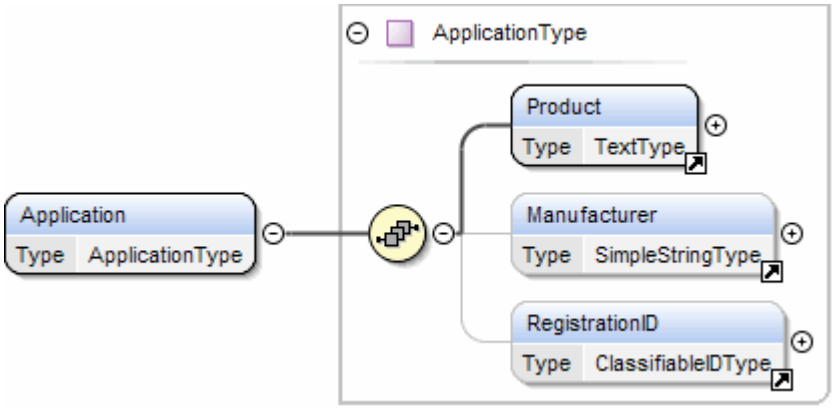
`xcpt:AnyXML` ist ein Erweiterungselement, das das Einfügen beliebigen XML-konformen Inhalts an definierten Stellen eines eXTra-Dokumentes erlaubt.

Beispiel

```
<AnyXML>
    <DatML-RAW-D xmlns:raw="http://www.destatis.de/schema/datml-raw/de/2.0" version="2.0">
        ...
    </DatML-RAW-D>
</AnyXML>
```

10.2 Element Application

	`xcpt:Application`
Inhalt	Bezeichnung einer Softwareanwendung
Datentyp	`##elements`
Benutzt von	`xcpt:RequestDetails`, `xcpt:ResponseDetails`



Beschreibung

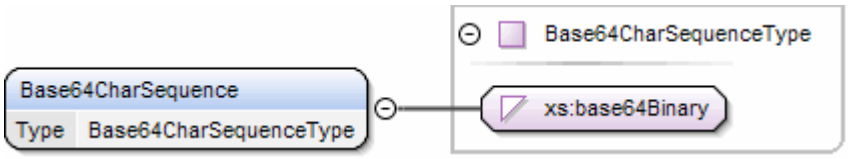
`xcpt:Application` dient dazu, die zur Erstellung einer eXTra-Nachricht verwendete Softwareanwendung wie folgt für Dokumentationszwecke zu beschreiben:

`Product`	Produktbezeichnung
`Manufacturer`	Hersteller
`RegistrationID`	Registrierungs-Id

Die Registierungs-Id dient zur Übertragung von Angaben mit denen eine Registrierung oder Zertifizierung der Anwendung dokumentiert wird.

10.3 Element Base64CharSequence

	xcpt:Base64CharSequence
Inhalt	Base64-Zeichenfolge
Datentyp	`xsd:base64Binary`
Benutzt von	`xcpt:DataType`



Attribute

	@numberOfChars	*optional*
Inhalt	Gibt die Anzahl der Zeichen der Base64-Zeichenfolge an	
Datentyp	`xsd:nonNegativeInteger`	

Beschreibung

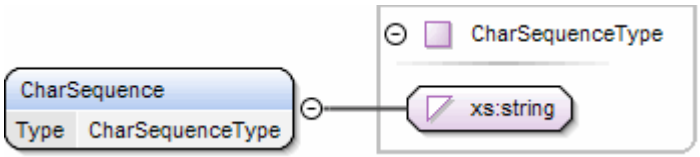
`xcpt:Base64CharSequence` beschränkt den Inhalt des Datencontainers auf eine Base64-konforme Zeichenfolge.

Hinweis:
Im Fall der Unterstützung zur Übermittlung großer Datenmengen im Gigabyte-Bereich gibt es diese Form ab der eXTra-Version 1.4 in Form der Unterstützung des *SOAP Transmision Optimization Mechanism* MTOM. MTOM kommt allerdings erst dann zum Tragen, wenn bei der Implementierung des Datenübermittlungssystems ein Framework zum Einsatz kommt, das ebenfalls MTOM unterstützt.

Bestehende Implementierungen von eXTra-spezifischen Datenübermittlungsverfahren, z.B. auf Basis der eXTra-Version 1.3, werden durch die Unterstützung von MTOM nicht beeinflusst. D.h. die entsprechende Erweiterung ist in der Schemadatei der components so erfolgt, dass sich bei der Verwendung dieser Schemadatei zur Validierung bestehender XML-Dokumenten der eXTra-Version 1.3 kein Änderungsbedarf ergibt.

10.4 Element CharSequence

	`xcpt:CharSequence`
Inhalt	XML-konforme Zeichenfolge
Datentyp	`xsd:string`
Benutzt von	`xcpt:DataType`



Attribute

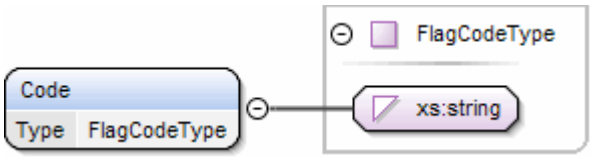
	`@numberOfChars`	*optional*
Inhalt	Gibt die Anzahl der Zeichen der Zeichenfolge an	
Datentyp	`xsd:nonNegativeInteger`	

Beschreibung

`xcpt:CharSequence` beschränkt den Inhalt des Datencontainers auf eine beliebige XML-konforme Zeichenfolge.

10.5 Element Code

	xcpt:Code
Inhalt	Ein anwendungsspezifischer Code
Datentyp	`xsd:string`
Benutzt von	`xcpt:Flag`

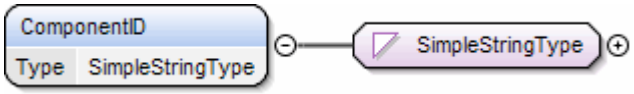


Beschreibung

`xcpt:Code` wird verwendet, um `xcpt:Flag` einen anwendungsspezifischen Code hinzuzufügen.

10.6 Element ComponentID

	`xcpt:ComponentID`
Inhalt	Anwendungsspezifische Referenz auf eine beliebige Komponente eines eXTra-Verarbeitungssystems
Datentyp	`xsd:string`
Benutzt von	`xlog:Event`, `xlog:Exception`, `xlog:LogSequence`, `xlog:LogStream`, `xlog:Message`, `xlog:Operation`

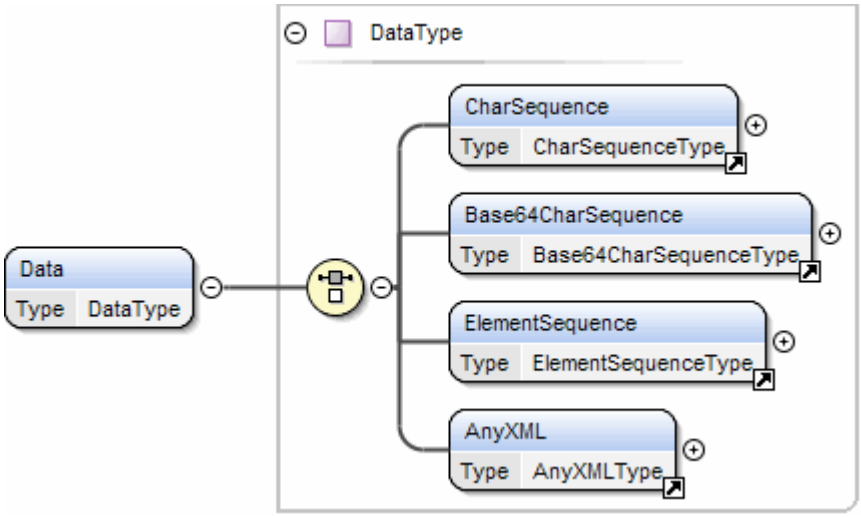


Beschreibung

`xcpt:ComponentID` dient dazu, in Logs auf beliebige Komponenten eines Verarbeitungssystems für eXTra-Nachrichten zu verweisen.

10.7 Element Data

	xcpt:Data
Inhalt	Eine *Typelement* mit anwendungsspezifischen Daten
Datentyp	`##elements`
Benutzt von	`xreq:MessageBody`, `xreq:PackageBody`, `xreq:TransportBody`, `xres:MessageBody`, `xres:PackageBody`, `xres:TransportBody`



Profilierung

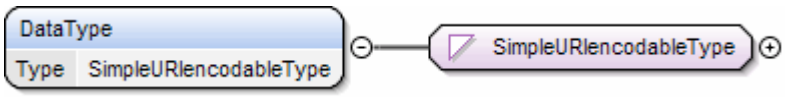
Komponente	*REQ*	*OPT*	*FOR*	*MUL*	*IGN*	*VAL*
`xcpt:CharSequence`	+	+	+		+	
`xcpt:Base64CharSequence`	+	+	+		+	
`xcpt:ElementSequence`	+	+	+		+	
`xcpt:AnyXML`	+	+	+		+	

Beschreibung

`xcpt:Data` ist das Wurzelement eines eXTra-Datencontainers. Es enthält genau eines von vier Typelementen, die den Datentyp der Nutznachricht formal eingrenzen und die anwendungsspezifischen Nutzdaten enthalten.

10.8 Element DataType

	xcpt:DataType
Inhalt	Art, Typ oder sonstige kategorisierende oder beschreibende anwendungs-spezifische Angabe über die übermittelten Daten.
Datentyp	`xsd:anyURI`
Benutzt von	`xcpt:RequestDetails`



Beschreibung

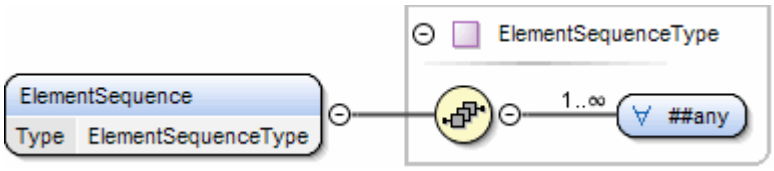
`xlog:DataType` dient zur Angabe einer Art-, Typ- oder sonstigen kategorisierenden oder beschreibenden Angabe über die übermittelten Daten. Ein häufiger Fall ist z.B. die formale Bezeichnung eines Datensatzes oder einer Datei.

Beispiel

```
<DataType>http://www.destatis.de/datatype/KU101/2008</DataType>
```

10.9 Element ElementSequence

	xcpt:ElementSequence
Inhalt	Eine Folge beliebiger XML-Elemente
Datentyp	`##elements`
Benutzt von	`xcpt:DataType`



Beschreibung

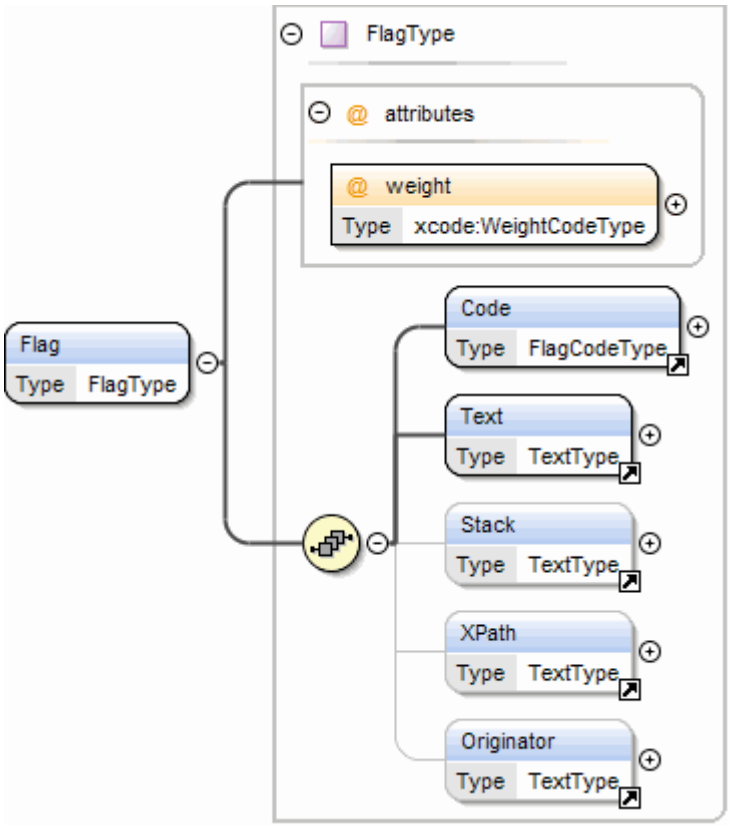
`xcpt:ElementSequence` ist eine der Typelemente des eXTra-Datencontainers; es beschränkt den Inhalt des Datencontainers auf eine Folge beliebiger XML-Elemente.

Beispiel

```
<ElementSequence>
    <DatML-RAW-D xmlns:raw="http://www.destatis.de/schema/datml-raw/de/2.0" version="2.0">
        ...
    </DatML-RAW-D>
</ElementSequence>
```

10.10 Element Flag

	`xcpt:Flag`
Inhalt	Beschreibt ein Meldungselement (Flag) eines Verarbeitungsprotokolls
Datentyp	`##elements`
Benutzt von	`xcpt:Report`



Attribute

	`@weight`	*optional*
Inhalt	Gewicht des Meldungselementes	
Datentyp	`xsd:anyURI`	
Werteraum	Codeliste *WeightCode*	

Beschreibung

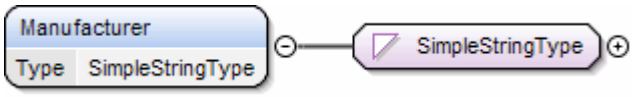
`xcpt:Flag` beschreibt ein einzelnes Meldungselement (Flag) eines Verarbeitungsprotokolls (`xcpt:Report`).

Beispiel

```
<Flag weight="http://www.extra-standard.de/weight/ERROR">
        <Code>X3301-500</Code>
        <Text>Http error 500 was raised</Text>
</Flag>
        <Flag weight="http://www.extra-standard.de/weight/ERROR">
        <Code>X4100-III</Code>
        <Text>Illegal value</Text>
        <XPath>XMLTransport/TransportHeader/RequestDetails/Scenario</XPath>
</Flag>
        <Flag weight="http://www.extra-standard.de/weight/INFO">
        <Code>X1000-2</Code>
        <Text>Server unavailable 2009-09-30T20:00P3H</Text>
</Flag>
```

10.11 Element Manufacturer

	xcpt:Manufacturer
Inhalt	Herstellerbezeichnung
Datentyp	`xsd:string`
Benutzt von	`xcpt:Application`



Attribute

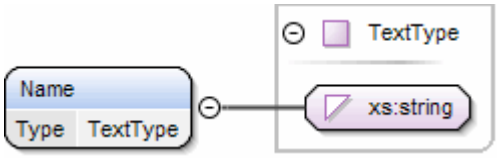
	@class	*optional*
Inhalt	Eine klassifizierende Angabe	
Datentyp	`xsd:anyURI`	

Beschreibung

Enthält den Namen oder eine andere Bezeichnung eines Herstellers.

10.12 Element Name

	`xcpt:Name`
Inhalt	Ein Name oder ein anderer beliebiger Bezeichner
Datentyp	`xsd:string`
Benutzt von	`xcpt:Receiver`, `xcpt:Sender`

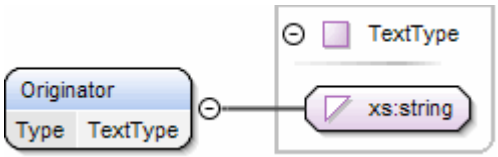


Beschreibung

Die Semantik des Namens ergibt sich aus dem Kontext, in dem das Element benutzt wird.

10.13 Element Originator

	xcpt:Originator
Inhalt	Angabe über einen Erzeuger/Verursacher
Datentyp	`xsd:string`
Benutzt von	`xcpt:Flag`



Beschreibung

Dieses Element wird für die Angabe eines Erzeugers bzw. Verursachers im Sinne einer technischen Komponente verwendet. In einem eXTra-Verarbeitungssystem ist dies typischerweise eine Soft- oder Hardwarekomponente.

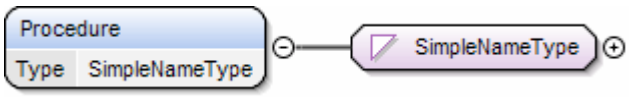
Eine Verwendung dieses Elementes ist beispielhaft in [BEST Kapitel 4.2.1., AcknowledgementUpdate] beschrieben. In diesem Anwendungsfalls wird Originator verwendet, um nach Auftreten eines Verarbeitungsfehler nach einem bereits bestätigten Empfang einen Bezug zur vorhergehende Response herzustellen.

Beispiel

```
<Originator>de.mycompany.extra.message.Validator</Originator>
```

10.14 Element Procedure

	xcpt:Procedure
Inhalt	Bezeichnung eines (softwaregestützten) technischen oder fachlichen Verfahrens
Datentyp	`xsd:string`
Benutzt von	`xcpt:RequestDetails`



Beschreibung

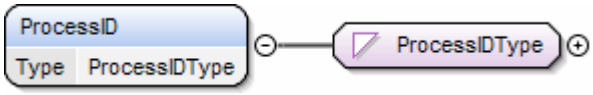
Mit diesem Element ist es möglich, das technische oder fachliche Verfahren, in dessen Kontext die Datenübermittlung stattfindet, zu dokumentieren.

Beispiel

```
<Procedure>http://www.destatis.de/procedures/eCORE#0122</Procedure>
```

10.15 Element ProcessID

	`xcpt:ProcessID`
Inhalt	Id eines technischen (IT-)Prozesses
Datentyp	`xsd:string`
Benutzt von	`xlog:LogSequence`, `xlog:LogStream`



Beschreibung

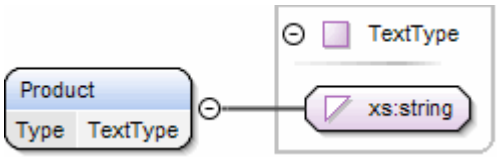
Dieses Element dient insbesondere im Kontext von Logs dazu, Instanzen von Verarbeitungsprozessen wie zum Beispiel Jobs oder Tasks zu dokumentieren.

Beispiel

```
<ProcessID>TSN:9A4F</ProcessID>
```

10.16 Element Product

	xcpt:Product
Inhalt	Bezeichnung eines (Software-) Produkts
Datentyp	`xsd:string`
Benutzt von	`xcpt:Application`



Beschreibung

Form und Inhalt der Produktbeschreibung sind beliebig.

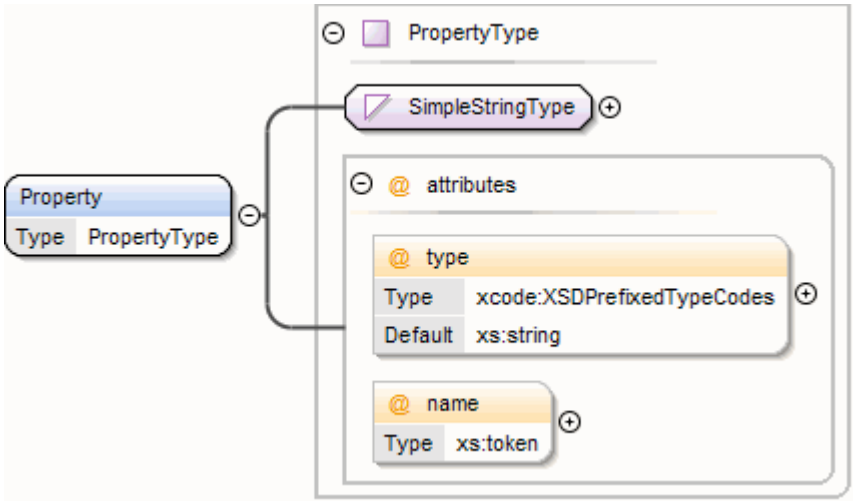
Beispiel

```
<Product>http://www.destatis.de/procducts/KonVert/1.6#Inspector</Product>

<Product>MySQL 5.1.34</Product>
```

10.17 Element Property

	xcpt:Property
Inhalt	Eigenschaft (zum Beispiel zur Zustandsbeschreibung)
Datentyp	`xsd:string`
Benutzt von	`xlog:State`



Beschreibung

`xcpt:Property` beschreibt eine Eigenschaft. Eine Eigenschaft hat einen Namen (erforderliches Attribut `name`) und einen Wert (Elementinhalt). Die Interpretation des Wertes kann durch die Angabe eines Datentyps unterstützt werden (Attribut `type`).

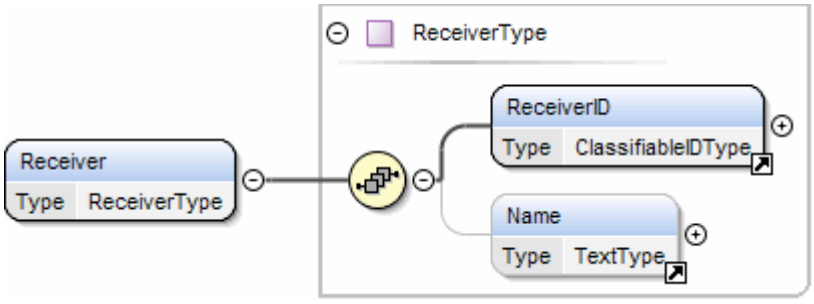
`xlog:State` verwendet `xcpt:Property`, um einen Zustand als eine Menge von Eigenschaften zu beschreiben.

Attribute

	@name	*erforderlich*
Inhalt	Name einer Eigenschaft	
Datentyp	`xsd:token`	
	@type	*optional*
Inhalt	Eine klassifizierende Angabe	
Datentyp	`xsd:token`	
Werteraum	`##codelist:XSDTypeCodes`	

10.18 Element Receiver

	xcpt:Receiver
Inhalt	Beschreibung eines Empfängers
Datentyp	`##elements`
Benutzt von	`xreq:MessageHeader`, `xreq:PackageHeader`, `xreq:TransportHeader`, `xres:MessageHeader`, `xres:PackageHeader`, `xres:TransportHeader`



Beschreibung

Ein Empfänger ist der Adressat einer Instanz eines Ebenenobjektes, also eines eXTra-Dokumentes, eines Paketes oder einer Nachricht. Es ist stets die formale, zwischen den

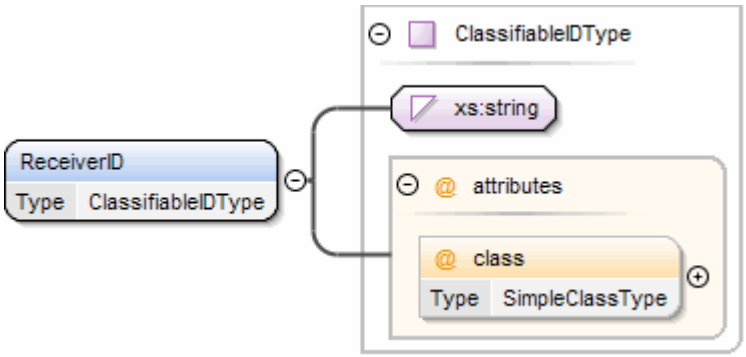
Kommunikationsteilnehmern vereinbarte bzw. bekannte Kennung des Empfängers anzugeben, da diese der eindeutigen Identifikation des Empfängers dient. Ein nicht-formaler Name kann optional ergänzt werden.

Beispiel

```
<Receiver>
        <ReceiverID>ID/TN/09S3F5BDA23</ReceiverID>
        <Name>SEND-IT Dienstleister für Datenkommunikation, Bitburg</Name>
</Receiver>
```

10.19 Element ReceiverID

	`xcpt:ReceiverID`
Inhalt	Formale Kennung zur Identifikation eines Empfängers
Datentyp	`xsd:string`
Benutzt von	`xcpt:Receiver`



Attribute

	`@class`	*optional*
Inhalt	Eine klassifizierende Angabe	
Datentyp	`xsd:anyURI`	

Beschreibung

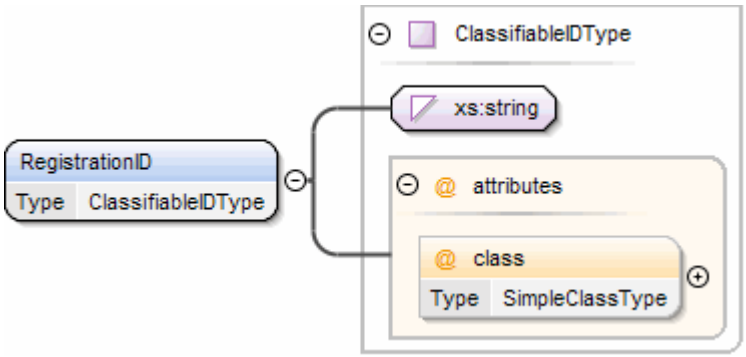
Siehe `xcpt:Receiver`.

Beispiel

```
<ReceiverID class="WZ">123456789</ReceiverID>
```

10.20 Element RegistrationID

	xcpt:RegistrationID
Inhalt	Kennung, mit der ein Objekt registriert oder zertifiziert wurde.
Datentyp	`xsd:string`
Benutzt von	`xcpt:Application`



Attribute

	@class	*optional*
Inhalt	Eine klassifizierende Angabe	
Datentyp	`xsd:anyURI`	

Beschreibung

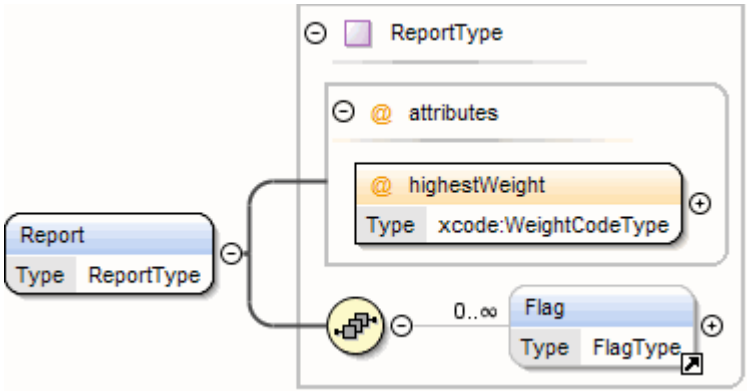
Siehe `xcpt:application`.

Beispiel

```
<RegistrationID class="DATEV">DSR20080926HID0412RID0034</RegistrationID>
```

10.21 Element Report

	`xcpt:Report`
Inhalt	Vom Empfänger erzeugter Bericht über die Verarbeitung eines Ebenenobjektes
Datentyp	`##elements`
Benutzt von	`xcpt:ResponseDetails`



Attribute

	`@highestWeight`	*optional*
Inhalt	Höchstes Gewicht der im Bericht enthaltenen Berichtselemente	
Datentyp	`xsd:anyURI`	
Werteraum	Codeliste *WeightCode*	

Beschreibung

Auf jeder Kommunikationsebene hat ein Empfänger die Möglichkeit, in einer Response für ein Ebenenobjekt einen Bericht über dessen Verarbeitung an den Absender zu übertragen. Der Bericht besteht aus einer Folge von Berichtselementen, sogenannten *Flags*, von denen jedes ein Gewicht mit den logischen Ausprägungen Information, Warnung und Fehler hat. Das Attribut `highestWeight` soll das höchste vergebene Gewicht anzeigen.

Beispiel

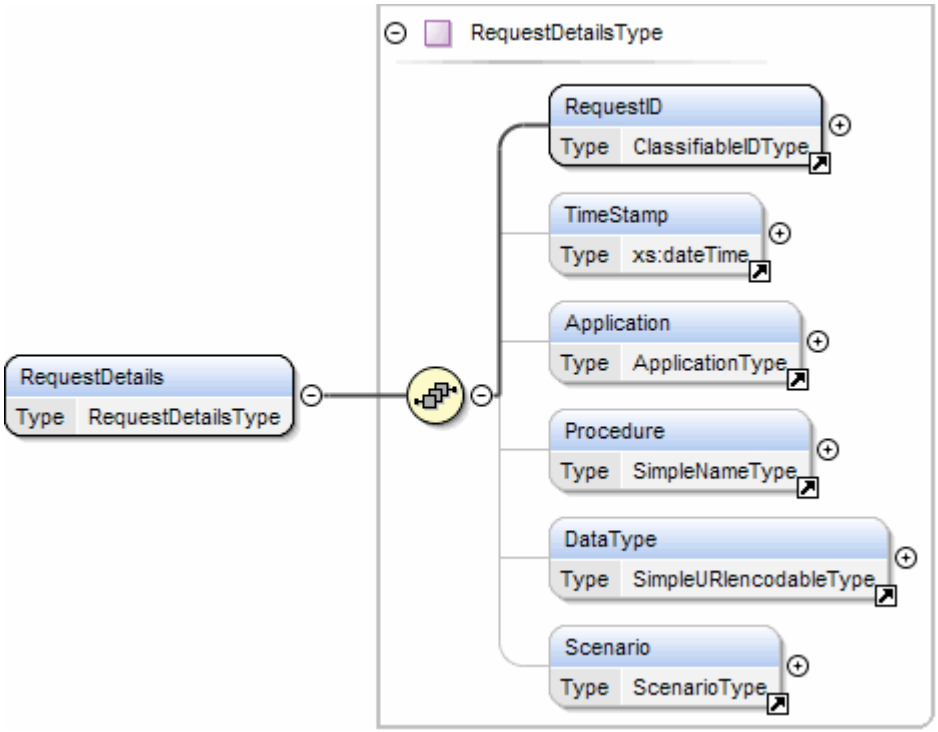
```
<Report highestWeight="http://www.extra-standard.de/weight/ERROR">
	<Flag weight="http://www.extra-standard.de/weight/ERROR">
		<Code>X3301-500</Code>
		<Text>Http error 500 was raised</Text>
	</Flag>

	<Flag weight="http://www.extra-standard.de/weight/ERROR">
		<Code>X4100-III</Code>
		<Text>Illegal value</Text>
		<XPath>XMLTransport/TransportHeader/RequestDetails/Scenario</XPath>
	</Flag>

	<Flag weight="http://www.extra-standard.de/weight/INFO">
		<Code>X1000-2</Code>
		<Text>Server unavailable 2009-09-30T20:00P3H</Text>
	</Flag>
</Report>
```

10.22 Element RequestDetails

	`xcpt:RequestDetails`
Inhalt	Diverse Request-spezifische Informationen
Datentyp	`##elements`
Benutzt von	`xreq:MessageHeader`, `xreq:PackageHeader`, `xreq:TransportHeader`, `xres:MessageHeader`, `xres:PackageHeader`, `xres:TransportHeader`



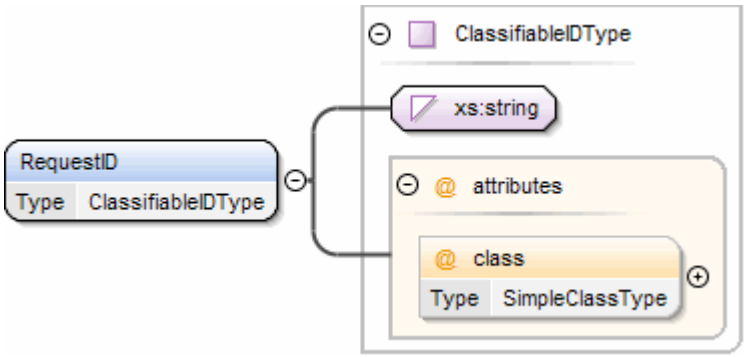
Beschreibung

Das Element beschreibt einen Request durch folgende Daten:

`RequestID`	Id des Request
`TimeStamp`	Zeitpunkt der Requesterzeugung
`Application`	Erzeugende Anwendung
`Procedure`	(Fachliches) Verfahren
`DataType`	Fachliche Bezeichnung der Daten
`Scenario`	Kommunikationsszenario

10.23 Element RequestID

	`xcpt:RequestID`
Inhalt	ID eines eXTra-Requests
Datentyp	`xsd:string`
Benutzt von	`xcpt:RequestDetails`



Attribute

	`@class`	*optional*
Inhalt	Eine klassifizierende Angabe	
Datentyp	`xsd:anyURI`	

Beschreibung

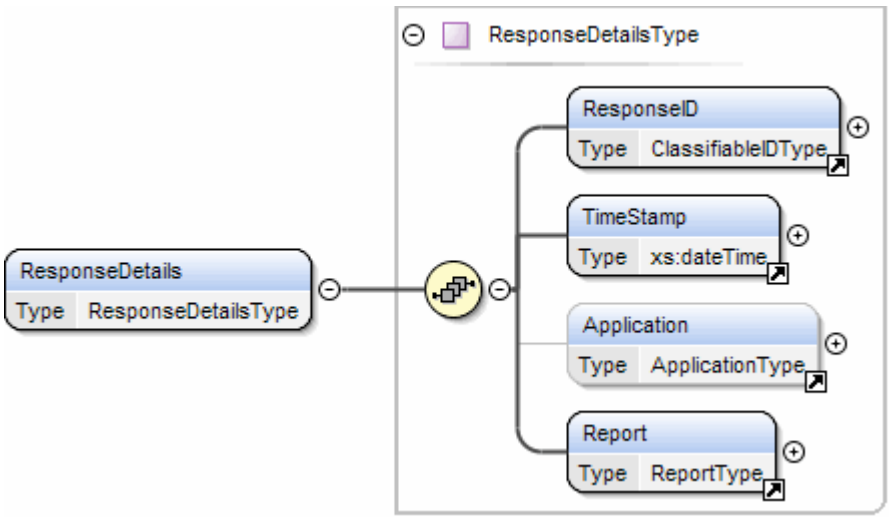
Siehe `xcpt:RequestDetails`.

Beispiel

```
<RequestID class="http//www.destatis.de/procedure/eCORE">REQ123456789</RequestID>
```

10.24 Element ResponseDetails

	`xcpt:ResponseDetails`
Inhalt	Diverse Response-spezifische Informationen
Datentyp	`##elements`
Benutzt von	`xres:MessageHeader`, `xres:PackageHeader`, `xres:TransportHeader`



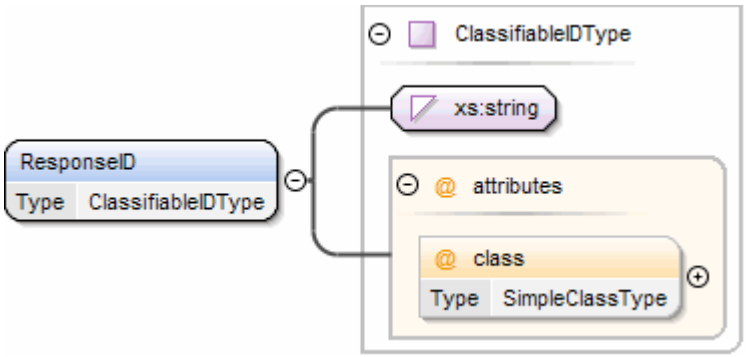
Beschreibung

Das Element beschreibt einen Response durch folgende Daten:

`ResponseID`	Id des Response
`TimeStamp`	Zeitpunkt der Requesterzeugung
`Application`	Erzeugende Anwendung
`Report`	Verarbeitungsbericht/-protokoll

10.25 Element ResponseID

	`xcpt:ResponseID`
Inhalt	ID eines eXTra-Response
Datentyp	`xsd:string`
Benutzt von	`xcpt:ResponseDetails`



Attribute

	`@class`	*optional*
Inhalt	Eine klassifizierende Angabe	
Datentyp	`xsd:anyURI`	

Beschreibung

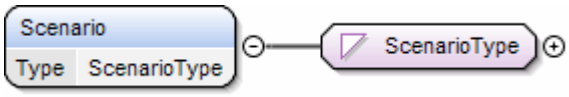
Siehe `xcpt:ResponseDetails.`

Beispiel

```
<ResponseID class="http//www.destatis.de/procedure/eCORE">RES123456789</ResponseID>
```

10.26 Element Scenario

	xcpt:Scenario
Inhalt	Spezifikation des Kommunikationsszenarios
Datentyp	Codeliste *ScenarioCode*
Benutzt von	`xcpt:RequestDetails`



Beschreibung

Gibt das von Absender gewünschte Kommunikationszenario an. Die Ausprägungen sind durch die Codeliste *ScenarioCode* festgelegt:

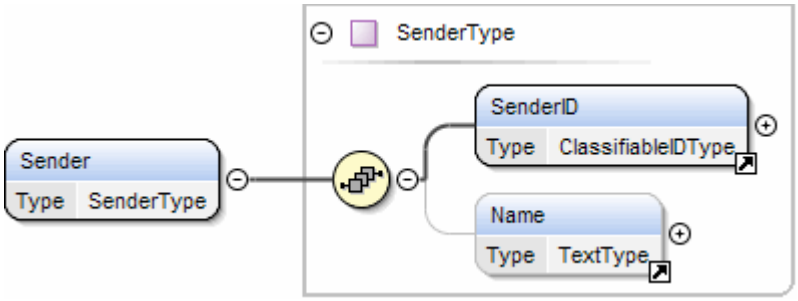
- `http://www.extra-standard.de/scenario/fire-and-forget`
- `http://www.extra-standard.de/scenario/request-with-response`
- `http://www.extra-standard.de/scenario/request-with-acknowledgement`

Beispiel

```
<Scenario>http://www.extra-standard.de/scenario/fire-and-forget</Scenario>
```

10.27 Element Sender

	xcpt:Sender
Inhalt	Beschreibung eines Absenders
Datentyp	`##elements`
Benutzt von	`xreq:MessageHeader, xreq:PackageHeader, xreq:TransportHeader, xres:MessageHeader, xres:PackageHeader, xres:TransportHeader`



Beschreibung

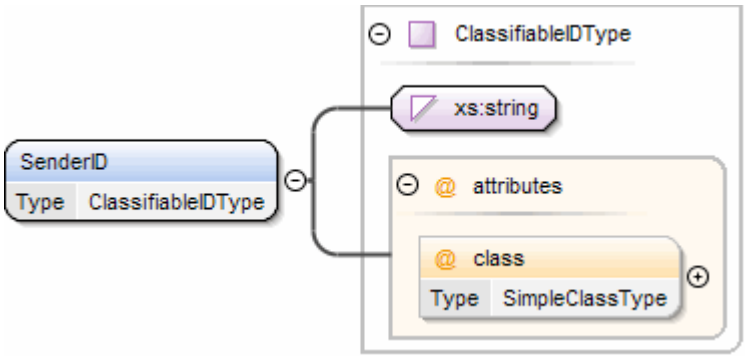
Ein Absender ist der Kommunikationspartner, der den Kommunikationsvorgang initiiert hat. Es ist stets die formale, zwischen den Kommunikationsteilnehmern vereinbarte bzw. bekannte Kennung des Absenders anzugeben, da diese der eindeutigen Identifikation dient. Ein nicht-formaler Name kann optional ergänzt werden.

Beispiel

```
<Sender>
        <SenderID>ID/TN/09S3F5BDA24</SenderID>
        <Name>SEND-IT Dienstleister für Datenkommunikation, Bitburg
        </Name>
</Sender>
```

10.28 Element SenderID

	`xcpt:SenderID`
Inhalt	Formale Kennung zur Identifikation eines Absenders
Datentyp	`xsd:string`
Benutzt von	`xcpt:Sender`



Attribute

	`@class`	*optional*
Inhalt	Eine klassifizierende Angabe	
Datentyp	`xsd:anyURI`	

Beschreibung

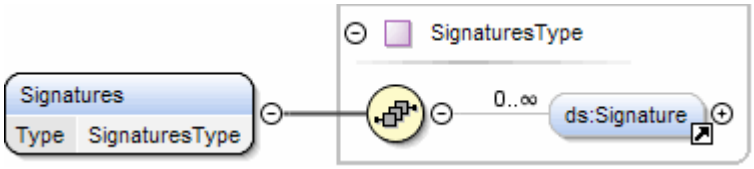
Siehe `xcpt:Sender`.

Beispiel

```
<SenderID>ID/TN/09S3F5BDA24</SenderID>
```

10.29 Element Signatures

	`xcpt:Signatures`
Inhalt	Eine beliebige Anzahl von *XML Signature* konformen Signaturen
Datentyp	`##elements`
Benutzt von	`xreq:Message`, `xreq:Package`, `xreq:XMLTransport`, `xres:Message`, `xres:Package`, `xres:XMLTransport`

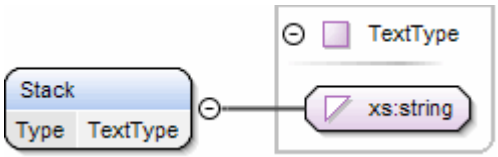


Beschreibung

xcpt:Signatures stellt einen Signaturblock dar. Für jede Instanz eines Ebenenobjektes steht ein separater Signaturblock zur Verfügung.

10.30 Element Stack

	xcpt:Stack
Inhalt	Ein Stackprint oder anderes Protokoll einer Aktion einer Anwendung
Datentyp	`xsd:string`
Benutzt von	`xcpt:Flag`

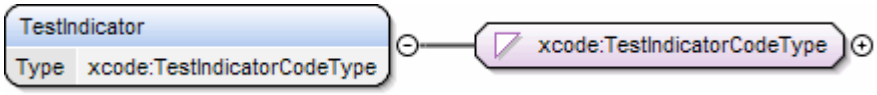


Beschreibung

Dieses Element wird benutzt, um Berichtselementen (Flags) Informationen über (insbesondere fehlerhafte) programmatische Abläufe oder Aktionen hinzuzufügen.

10.31 Element TestIndicator

	xcpt:Testindicator
Inhalt	Indikator für Testfälle und Testfallverarbeitung
Datentyp	Codeliste *TestIndicatorCode*
Benutzt von	xreq:MessageHeader, xreq:PackageHeader, xreq:TransportHeader, xres:MessageHeader, xres:PackageHeader, xres:TransportHeader



Beschreibung

Mit diesem Element wird beschrieben, ob das betroffene Ebenenobjekt Teststatus hat, und wie Ebenenobjekte mit Teststatus weiterzuverarbeiten sind. Bei Abwesenheit des Elementes wird angenommen, dass das Ebenenobjekt Produktivstatus hat.

Die möglichen Ausprägungen des Inhaltes sind durch die Codeliste *TestIndicatorCode* festgelegt.

```
http://www.extra-standard.de/test/NONE
http://www.extra-standard.de/test/RECEIVE
http://www.extra-standard.de/test/ACCEPT
http://www.extra-standard.de/test/PROCESS
```

Die Bedeutung der verschiedenen Ausprägungen ist wie folgt:

- `RECEIVE`: Test des Übertragungsvorgangs, einschließlich Validierung des erhaltenen eXTra-Dokumentes gegenüber dem profilierten Schema; die empfangenen Daten werden ignoriert.
  Dieser Testmerker eignet sich u.a. für die ersten Tests im Rahmen der Realisierung und Inbetriebnahme eines neuen Datenübermittlungssystems, um die formale Korrektheit der übermittelten eXTra-Dokumente zu verifizieren.
- `ACCEPT`: Test des Übertragungsvorgangs und der bei der Entgegennahme der Daten notwendigen Arbeitsschritte, wie z.B. Validierung des erhaltenen eXTra-Dokumentes gegenüber dem profilierten Schema, Übernahme in die lokale Datenhaltung, Komprimieren/Dekomprimieren, Verschlüsseln/Entschlüsseln, Signieren/Signatur prüfen usw. Die Daten werden nicht weitergereicht.
- `PROCESS`: Weiterreichen der als Testdaten gekennzeichneten Daten an die nächste Ebene bzw. an das Fachverfahren. Diese Möglichkeit kann dann sinnvoll sein, wenn

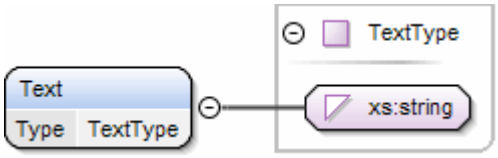
als Folge der Verarbeitung weitere asynchron abzuholende Informationen, wie z.B. Protokolle oder Bescheide, erstellt werden, die ja ebenfalls getestet werden sollten.

Beispiel

```
<TestIndicator>http.//www.extra-standard.de/test/RECEIVE</TestIndicator>
```

10.32 Element Text

	xcpt:Text
Inhalt	Ein beliebiger Text
Datentyp	`xsd:string`
Benutzt von	`xcpt:Flag`, `xlog:Message`, `xlog:StrackTrace`



Beschreibung

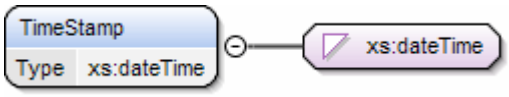
`xcpt:Text` wird in der Struktur des eXTra-Protokolls verwendet, um Texte als ergänzende oder zentrale Information zu übertragen, z.B. in Logs.

Beispiel

```
<Text>Der Wert des Elementes 'Scenario' ist unzulaessig</Text>
```

10.33 Element TimeStamp

	xcpt:TimeStamp
Inhalt	Ein Zeitstempel
Datentyp	`xsd:dateTime`
Benutzt von	`xcpt:RequestDetails`, `xcpt:ResponseDetails`, `xlog:Event`, `xlog:Exception`, `xlog:LogSequence`, `xlog:LogStream`, `xlog:Message`, `xlog:Operation`, `xlog:State`



Beschreibung

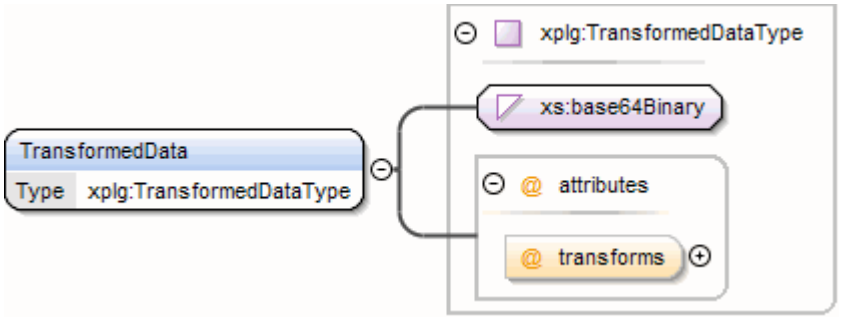
Zeitstempel sind für die Dokumentation einer Datenübermittlung von zentraler Bedeutung und daher an vielen Stellen des eXTra-Protokolls Pflicht.

Beispiel

```
<TimeStamp>2013-09-26T06:09:26</TimeStamp>
```

10.34 Element TransformedData

	`xcpt:TransformedData`
Inhalt	Daten, die einer Transformation wie z.B. Verschlüsselung unterzogen wurden.
Datentyp	`xsd:base64Binary`
Benutzt von	`xreq:MessageBody`, `xreq:PackageBody`, `xreq:TransportBody`, `xres:MessageBody`, `xres:PackageBody`, `xres:TransportBody`

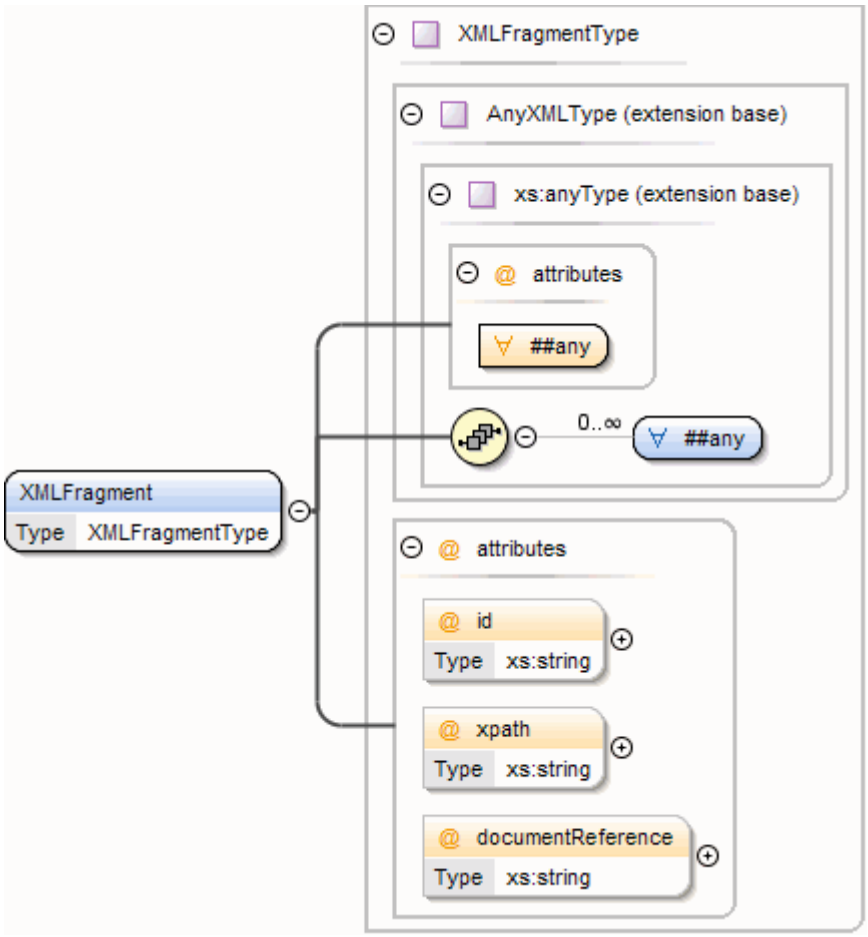


Beschreibung

`xcpt:TransformedData` enthält Daten, die einer oder mehreren Transformationen wie z.B. Verschlüsselung und Komprimierung unterzogen wurden. Durch Umkehrung der Transformationen werden die Originaldaten wiederhergestellt und können anstelle von xcpt:TransformedData in den Datenstrom eingefügt werden. Das Attribut `transforms` erlaubt es, Transformationen in der Reihenfolge ihrer Anwendung aufzulisten. Dazu muss jede Transformation in einem `DataTransforms` Plug-In beschrieben sein.

10.35 Element XMLFragment

	`xcpt:XMLFragment`
Inhalt	Ein XML Fragment in einem beliebigen Namensraum
Datentyp	`##elements`
Benutzt von	`xlog:State`



Attribute

	`@id`	*optional*
Inhalt	Eine beliebige ID	
Datentyp	`xsd:string`	

	`@xpath`	*optional*
Inhalt	Ein XPath Ausdruck	
Datentyp	`xsd:string`	

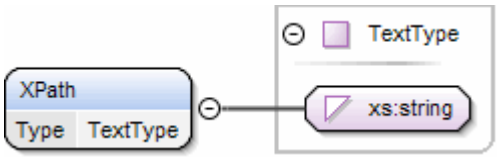
	`@documentReference`	*optional*
Inhalt	Eine Verweis auf ein (XML) Dokument	
Datentyp	`xsd:string`	

Beschreibung

`xcpt:XMLFragment` erlaubt das Einfügen anwendungsspezifischer XML-Strukturen an dedizierten Stellen des eXTra-Protokolls. Die Verwendung bzw. Semantik ergibt sich aus dem Kontext der Verwendung.

10.36 Element XPath

	xcpt:XPath
Inhalt	Ein XPath Ausdruck
Datentyp	`xsd:string`
Benutzt von	`xcpt:Flag`



Beschreibung

Mit einem XPath Ausdruck wird ein Knoten in einem XML Dokument adressiert. `xcpt:Flag` benutzt `xcpt:XPath` um auf eine mit dem Flag in Zusammenhang stehende Position im eXTra-Dokument zu verweisen.

11 Ressourcen

11.1 XSD Schema Ressourcen

Das XSD Schema der eXTra-Basisschnittstelle besteht aus den folgenden Schemadateien.

`eXTra-request-1.xsd`	eXTra-Nachricht von Typ Request
`eXTra-response-1.xsd`	eXTra-Nachricht von Typ Response
`eXTra-components-1.xsd`	Allgemeine Komponenten
`eXTra-codelists-1.xsd`	Kodelisten
`eXTra-logging-1.xsd`	Logging-Komponente
`eXTra-plugins-1.xsd`	PlugIns
`xenc-schema.xsd`	Komponente XML Encryption
`xmldsig-core-schema.xsd`	Komponente XML Signature
`eXTra-error-1.xsd`	Fehlernachricht

12 Literatur

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/