



*einheitliches XML-basiertes Transportverfahren*

**eXTra Basis-Standard**

**eXTra-Transport V1.2**

**Schnittstellenbeschreibung  
Ausgabestand 1.2.1**

FINAL

**Herausgeber:**

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

## 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.<br>Neue Nachricht <code>ExtraError</code><br>Neues PlugIn <code>Contact</code><br>PlugIn <code>DataTransforms</code> : Der Signatur kann jetzt das Zertifikat beigelegt werden..<br>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

|  |    |
|--|----|
| Einleitung und Übersicht.....                        | 7  |
| Verwendung dieses Dokumentes .....                   | 10 |
| Layout und Notationen .....                          | 11 |
| Formatierungen .....                                 | 11 |
| Beschreibung der Regeln .....                        | 11 |
| Beschreibung der Elemente und Attribute .....        | 11 |
| Graphische Symbole .....                             | 13 |
| Definitionen [normativ].....                         | 16 |
| Begriffe.....  | 16 |
| Namensräume und Präfixe .....                        | 16 |
| Aufbau von eXtra Nachrichten [normativ] .....        | 18 |
| Versionierung von eXtra Komponenten [normativ] ..... | 19 |
| eXtra Codelisten [normativ] .....                    | 21 |
| Codeliste CompressionAlgorithmCode .....             | 22 |
| Codeliste DataContainerCode .....                    | 22 |
| Codeliste EncryptionAlgorithmCode .....              | 22 |
| Codeliste OperationCode .....                        | 23 |
| Codeliste ScenarioCode .....                         | 23 |
| Codeliste SignatureAlgorithmCode .....               | 23 |
| Codeliste TestIndicatorCode .....                    | 24 |
| Codeliste UsageCode.....                             | 24 |
| Codeliste WeightCode .....                           | 24 |
| Codeliste XSDPrefixedType .....                      | 25 |
| Nachrichtentyp Request [normativ].....               | 26 |
| Element XMLTransport.....                            | 27 |
| Element Message.....                                 | 29 |
| Element MessageBody.....                             | 31 |
| Element MessageHeader .....                          | 32 |
| Element MessagePlugIns .....                         | 33 |
| Element Package .....                                | 34 |
| Element PackageBody .....                            | 36 |
| Element PackageHeader.....                           | 38 |
| Element PackagePlugIns.....                          | 39 |
| Element TransportBody.....                           | 40 |
| Element TransportHeader .....                        | 42 |
| Element TransportPlugIns .....                       | 43 |
| Nachrichtentyp Response [normativ] .....             | 44 |
| Element XMLTransport.....                            | 45 |
| Element Message.....                                 | 47 |
| Element MessageBody.....                             | 49 |
| Element MessageHeader .....                          | 50 |
| Element MessagePlugIns .....                         | 52 |
| Element Package .....                                | 53 |
| Element PackageBody .....                            | 55 |
| Element PackageHeader.....                           | 57 |
| Element PackagePlugIns.....                          | 58 |
| Element TransportBody .....                          | 59 |
| Element TransportHeader .....                        | 61 |
| Element TransportPlugIns .....                       | 62 |
| Extra Error [normativ] .....                         | 63 |
| Element ExtraError .....                             | 64 |

|   |     |
|---|-----|
| Element Reason .....                    | 65  |
| Plug-Ins [normativ] .....               | 66  |
| Element Value .....                     | 67  |
| Element X509Certificate .....           | 68  |
| Plug-In Certificates .....              | 69  |
| Element X509EncCertificate .....        | 70  |
| Plug-In Contacts .....                  | 71  |
| Element SenderContact .....             | 72  |
| Plug-In DataSource .....                | 73  |
| Element DataContainer .....             | 74  |
| Element DataSet .....                   | 76  |
| Plug-In DataTransforms .....            | 77  |
| Element Compression .....               | 79  |
| Element Compression/Algorithm .....     | 80  |
| Element Encryption .....                | 82  |
| Element Encryption/Algorithm .....      | 83  |
| Element InputData .....                 | 85  |
| Element OutputData .....                | 86  |
| Element Signature .....                 | 87  |
| Element Signature/Algorithm .....       | 89  |
| Element Specification .....             | 91  |
| Logging [normativ] .....                | 92  |
| Element Logging .....                   | 93  |
| Element Event .....                     | 95  |
| Element Exception .....                 | 96  |
| Element LogID .....                     | 97  |
| Element LogSequence .....               | 98  |
| Element LogStream .....                 | 100 |
| Element LogTrace .....                  | 101 |
| Element Message .....                   | 102 |
| Element Object .....                    | 103 |
| Element Operation .....                 | 105 |
| Element Parameter .....                 | 107 |
| Element PreviousLogID .....             | 109 |
| Element StackTrace .....                | 110 |
| Element State .....                     | 111 |
| Allgemeine Komponenten [normativ] ..... | 112 |
| Element AnyXML .....                    | 113 |
| Element Application .....               | 114 |
| Element Base64CharSequence .....        | 115 |
| Element CharSequence .....              | 116 |
| Element Code .....                      | 117 |
| Element ComponentID .....               | 118 |
| Element Data .....                      | 119 |
| Element DataType .....                  | 120 |
| Element ElementSequence .....           | 121 |
| Element Flag .....                      | 122 |
| Element Manufacturer .....              | 124 |
| Element Name .....                      | 125 |
| Element Originator .....                | 126 |
| Element Procedure .....                 | 127 |
| Element ProcessID .....                 | 128 |
| Element Product .....                   | 129 |

|                               |     |
|-------------------------------|-----|
| Element Property .....        | 129 |
| Element Receiver .....        | 130 |
| Element ReceiverID .....      | 132 |
| Element RegistrationID .....  | 133 |
| Element Report .....          | 134 |
| Element RequestDetails .....  | 136 |
| Element RequestID .....       | 138 |
| Element ResponseDetails ..... | 139 |
| Element ResponseID .....      | 140 |
| Element Scenario .....        | 141 |
| Element Sender .....          | 142 |
| Element SenderID .....        | 143 |
| Element Signatures .....      | 144 |
| Element Stack .....           | 145 |
| Element TestIndicator .....   | 146 |
| Element Text .....            | 148 |
| Element TimeStamp .....       | 149 |
| Element TransformedData ..... | 150 |
| Element XMLFragment .....     | 151 |
| Element XPath .....           | 153 |
| Ressourcen .....              | 154 |
| XSD Schema Ressourcen .....   | 154 |
| Literatur .....               | 155 |

## Einleitung und Übersicht

*Gültigkeit und Versionierung:* Dieses Dokument mit dem Ausgabestand 1.2.1 enthält die normative technische Spezifikation der Version 1.2 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 (hier Version 1.2) der zugrundeliegenden Transport-Schnittstelle des eXTra Basis Standards angeführt, während die letzte Ziffer r den jeweiligen Ausgabestand (hier Ausgabestand 1) wiedergibt.

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

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

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 im Implementierungseitfaden [IMPL] 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. Die Profilierung der eXTra Kommunikationsschnittstelle ist in [PROF] erläutert.

Der eXTra Standard umfasst also neben der in diesem Dokument beschriebenen Transportschicht eine Reihe sogenannter eXTra Standardnachrichten. Standardnachrichten sind als Nutzdaten zu betrachten und wie diese zu verarbeiten. Aus diesem Grund befinden sich in einem separaten Namensraum und haben keinerlei Abhängigkeit zu den Schemata der eXTra Transportschicht.

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 des eXTra Standards 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 Definition – 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 Teil der Schnittstellenimplementierung zu betrachten, welche vollständig nach außen gekapselt ist. Jedes Schema, das die Schnittstellenspezifikation korrekt implementiert, muss jede Nachrichteninstanz, die alle Regeln der Schnittstellenspezifikation erfüllt, als valide erkennen.

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

Der normative Spezifikationsteil beginnt mit den einleitenden Kapiteln, Definitionen und Aufbau von eXTra Nachrichten. Anschließend folgt die Spezifikation den Namensräumen [XNAM] der Elemente. Jeder Namensraum ist in einem separaten Kapitel beschrieben. Soweit in einem Namensraum ein Wurzelement 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.

## **Verwendung dieses Dokumentes**

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.

## Layout und Notationen

### 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.

### 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                | <i>Implementierung</i>   |
| P                | <i>Profilierung</i>      |
| X                | <i>XML Schnittstelle</i> |

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 der Regelbereich, rg die Regelgruppe und rn die Regelnummer angibt. Beispiele:

[I 12], [P 8.3].

### 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“

|                    | <i>präfigierter_elementname</i>         |
|--------------------|---|
| <i>Inhalt</i>      | <i>kurzbeschreibung_element</i>         |
| <i>Datentyp</i>    | <i>inhaltstyp_element</i>               |
| <i>Benutzt von</i> | <i>präfigierte_namen_elternelemente</i> |
|                    | <i>element_graphik</i>                  |

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“

|                 | <i>Attributname</i>              | <i>optionalität</i> |
|-----------------|----------------------------------|---------------------|
| <i>Inhalt</i>   | <i>kurzbeschreibung_attribut</i> |                     |
| <i>Datentyp</i> | <i>xsd-datentyp</i>              |                     |

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“**

| <i>Komponente</i>      | <i>REQ</i> | <i>OPT</i> | <i>FOR</i>      | <i>MUL</i> | <i>IGN</i> | <i>VAL</i> |
|------------------------|------------|------------|-----------------|------------|------------|------------|
| <b>xcpt:Signatures</b> | +          | +          | +               |            | +          |            |
| <b>xcpt:Data</b>       |            |            | <i>IMPLICIT</i> |            |            |            |

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

Die Bezeichner bedeuten:

- *REQ*: Profilierung kann die Komponente als erforderlich (required) deklarieren
- *OPT*: Profilierung kann die Komponente als optional (optional) deklarieren
- *FOR*: Profilierung kann die Komponente als verboten (forbidden) deklarieren
- *MUL*: Profilierung kann die Komponente als m bis n mal vorkommend deklarieren
- *IGN*: Profilierung kann die Komponente als ignoriert (ignored) deklarieren
- *VAL*: Der Wert (value) ist profilierbar (nur Elemente mit Dateninhalt und Attribute)

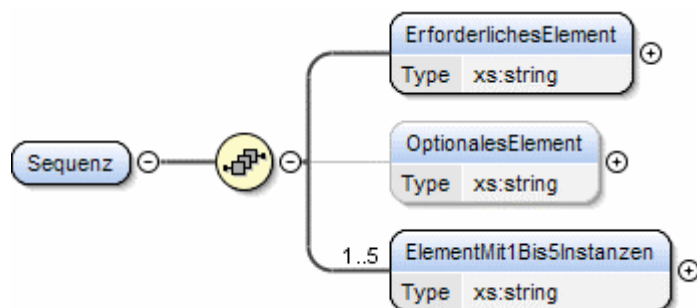
## **Graphische Symbole**

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

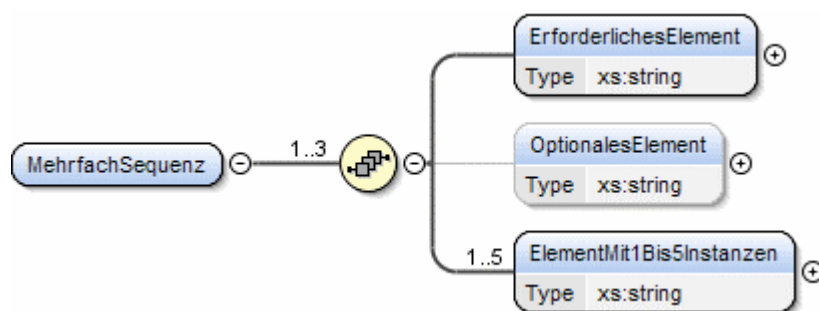
*Hinweis*: In den meisten Fällen erscheinen Elemente und Attribute in einem im Farbruck gelb eingefärbten Rechteck. Dieses stellt eine XSD-Typdefinition dar und trägt deren Namen. Typdefinitionen sind schemaspezifisch und daher nicht normativ. Aus diesem Grund findet sich in den folgenden Mustergraphiken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

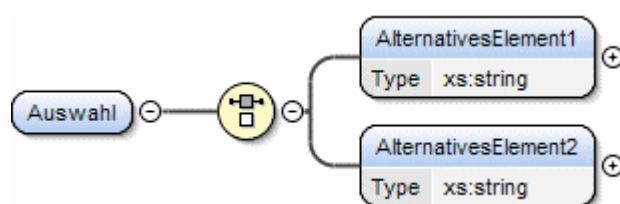
### Eine Sequenz von Elementen



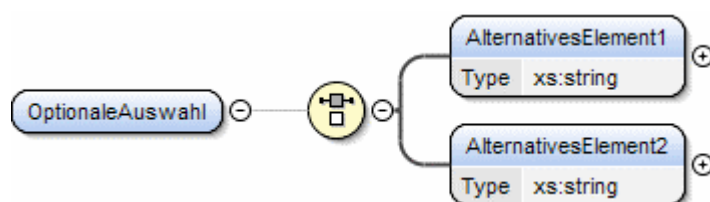
### Eine bis zu 3 mal wiederholte Sequenz von Elementen



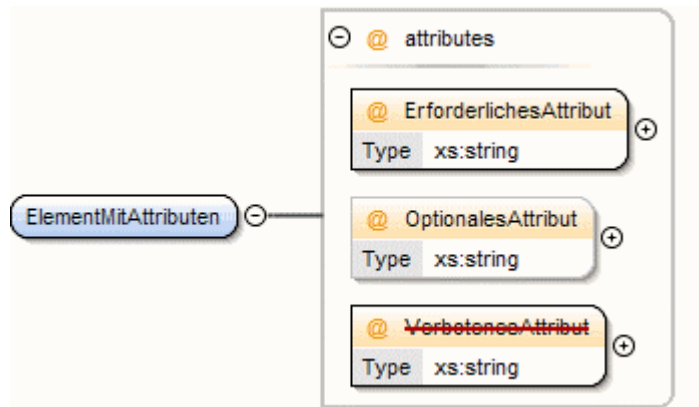
### Eine erforderliche Auswahl (Kindelemente sind Pflichtelemente)



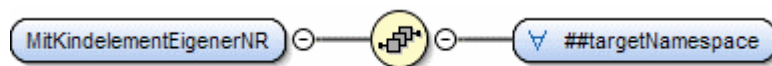
### Eine optionale Auswahl von (Kindelemente sind Pflichtelemente)



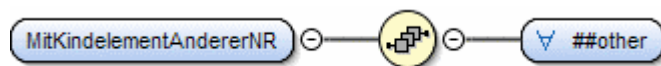
## Ein Element mit Attributen



## Ein Element mit einem beliebigem Kindelement aus dem eigenen Namensraum



## Ein Element mit einem beliebigem Kindelement aus einem anderen Namensraum



## Definitionen [normativ]

### ***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 Kapitel in allen konjugierten bzw. deklinierten Formen erscheinen.

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

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

### ***Namensräume und Präfixe***

Der Basisstandard 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.

| <b>Namensraum</b>   | <b>Präfix</b> |
|---|---------------|
| <a href="http://www.extra-standard.de/namespace/request/1">http://www.extra-standard.de/namespace/request/1</a>       | xreq, xtra    |
| <a href="http://www.extra-standard.de/namespace/response/1">http://www.extra-standard.de/namespace/response/1</a>     | xres, xtra    |
| <a href="http://www.extra-standard.de/namespace/components/1">http://www.extra-standard.de/namespace/components/1</a> | xcpt          |





**eXTra Transport V1.2**  
**Schnittstellenbeschreibung V1.2.1**

Seite: 17  
Status: **FINAL**  
Stand: **08.09.2010**

<http://www.extra-standard.de/namespace/plugin/1>  
<http://www.extra-standard.de/namespace/logging/1>  
<http://www.extra-standard.de/namespace/service/1>  
<http://www.w3.org/2001/04/xmlenc#>  
<http://www.w3.org/2000/09/xmlsig#>

xplg  
xlog  
xsrv  
xenc  
ds

## **Aufbau von eXtra Nachrichten [normativ]**

Hier nur Grundsätzliches, Zusammenhänge zwischen Namensräumen etc. Die Details sollten in [IMPL] stehen.

## Versionierung von eXTra Komponenten [normativ]

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.

| <b>eXtra Release</b>        | <b>1.0</b> | <b>1.1</b>     | <b>1.2</b> |
|-----------------------------|------------|----------------|------------|
| <b><i>Basisstandard</i></b> |            |                |            |
| Request                     | 1.0        | 1.1            | 1.2        |
| Response                    | 1.0        | 1.1            | 1.2        |
| Logging                     | 1.0        | 1.1            | 1.2        |
| Codelists                   | 1.0        | 1.1            | 1.2        |
| Components                  | 1.0        | 1.1            | 1.2        |
| Error                       | -          | -              | 1.0        |
| <b><i>Plug-Ins</i></b>      |            |                |            |
| Certificates                | -          | 1.0            | 1.0        |
| Communication               | 1.0        | <del>1.0</del> | -          |
| Contacts                    | -          | -              | 1.0        |
| DataSource                  | 1.0        | 1.0            | 1.0        |
| DataTransforms              | 1.0        | 1.1            | 1.2        |

### ***Versionsmatrix Standardnachrichten***

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

| <b>eXtra Release</b>              | <b>1.0</b> | <b>1.1</b> | <b>1.2</b> |
|-----------------------------------|------------|------------|------------|
| <b><i>Standardnachrichten</i></b> |            |            |            |
| ConfirmationOfReceipt             | 1.0        | 1.1        | 1.2        |
| DataRequest                       | 1.0        | 1.1        | 1.1        |
| ListOfConfirmationOfReceipt       | -          | -          | 1.0        |
| ListOfDataRequest                 | -          | -          | 1.0        |

## 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. Es 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**

|        |                        |
|--------|------------------------|
| #0     | Nur Ziffern erlaubt    |
| #noSig | Nur Buchstaben erlaubt |
| #A-1.B | Gültige Unterteilung   |

### **Ungültige Ausdrücke**

|         |                          |
|---------|--------------------------|
| 0       | ‚#‘ fehlt                |
| #no/Sig | Unzulässiges Zeichen ‚/‘ |
| #.A--1  | Ungültige Unterteilung   |

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

## ***Codeliste CompressionAlgorithmCode***

|                           |   |
|---------------------------|---|
|                           | <b>xcode:CompressionAlgorithmCodeType</b>     |
| <i>Inhalt</i>             | Identifikatoren für Komprimierungsalgorithmen |
| <i>Datentyp</i>           | xsd:anyURI                                    |
| <i>Individualisierbar</i> | Ja  |
| <i>Beliebige Domain</i>   | 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>

## ***Codeliste DataContainerCode***

|                           |                                    |
|---------------------------|------------------------------------|
|                           | <b>xcode:DataContainerCodeType</b> |
| <i>Inhalt</i>             | Identifikatoren für Containertypen |
| <i>Datentyp</i>           | xsd:anyURI                         |
| <i>Individualisierbar</i> | Ja                                 |
| <i>Beliebige Domain</i>   | Nein                               |

### *Vordefinierte Werte:*

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

## ***Codeliste EncryptionAlgorithmCode***

|                           |   |
|---------------------------|---|
|                           | <b>xcode:EncryptionAlgorithmCodeType</b>        |
| <i>Inhalt</i>             | Identifikatoren für Verschlüsselungsalgorithmen |
| <i>Datentyp</i>           | xsd:anyURI                                      |
| <i>Individualisierbar</i> | Ja  |
| <i>Beliebige Domain</i>   | 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>

## ***Codeliste OperationCode***

**xcode:OperationCodeType**

|                           |   |
|---------------------------|---|
| <i>Inhalt</i>             | Identifikatoren für Beschreibung von Operationen im Log |
| <i>Datentyp</i>           | xsd:anyURI  |
| <i>Individualisierbar</i> | Ja  |
| <i>Beliebige Domain</i>   | 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>

## ***Codeliste ScenarioCode***

**xcode:ScenarioCodeType**

|                           |   |
|---------------------------|---|
| <i>Inhalt</i>             | Identifikatoren für das Kommunikationszenario |
| <i>Datentyp</i>           | xsd:anyURI                                    |
| <i>Individualisierbar</i> | Ja  |
| <i>Beliebige Domain</i>   | 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>

## ***Codeliste SignatureAlgorithmCode***

**xcode:SignatureAlgorithmType**

|                           |                                       |
|---------------------------|---------------------------------------|
| <i>Inhalt</i>             | Identifikatoren für Signaturverfahren |
| <i>Datentyp</i>           | xsd:anyURI                            |
| <i>Individualisierbar</i> | Ja                                    |
| <i>Beliebige Domain</i>   | 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>

## ***Codeliste TestIndicatorCode***

|                           |   |
|---------------------------|---|
|                           | <b>xcode:TestIndicatorCodeType</b>                            |
| <i>Inhalt</i>             | Identifikatoren für Erkennung und Verarbeitung von Testfällen |
| <i>Datentyp</i>           | xsd:anyURI  |
| <i>Individualisierbar</i> | Ja  |
| <i>Beliebige Domain</i>   | 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>

## ***Codeliste UsageCode***

|                           |  |
|---------------------------|--|
|                           | <b>xcode:UsageCodeType</b>                                     |
| <i>Inhalt</i>             | Identifikatoren die Verwendung im Log dokumentierter Parameter |
| <i>Datentyp</i>           | xsd:anyURI   |
| <i>Individualisierbar</i> | Ja   |
| <i>Beliebige Domain</i>   | 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>

## ***Codeliste WeightCode***

|                           |                                    |
|---------------------------|------------------------------------|
|                           | <b>xcode:WeightCodeType</b>        |
| <i>Inhalt</i>             | Identifikatoren für Fehlergewichte |
| <i>Datentyp</i>           | xsd:anyURI                         |
| <i>Individualisierbar</i> | Ja                                 |
| <i>Beliebige Domain</i>   | 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>

## ***Codeliste XSDPrefixedType***

|                 |   |
|-----------------|---|
|                 | <b>xcode:XSDPrefixedTypeCodes</b>             |
| <i>Inhalt</i>   | Liste von XSD Datentypen in präfigierter Form |
| <i>Datentyp</i> | xsd:QName                                     |

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

## **Nachrichtentyp Request [normativ]**

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

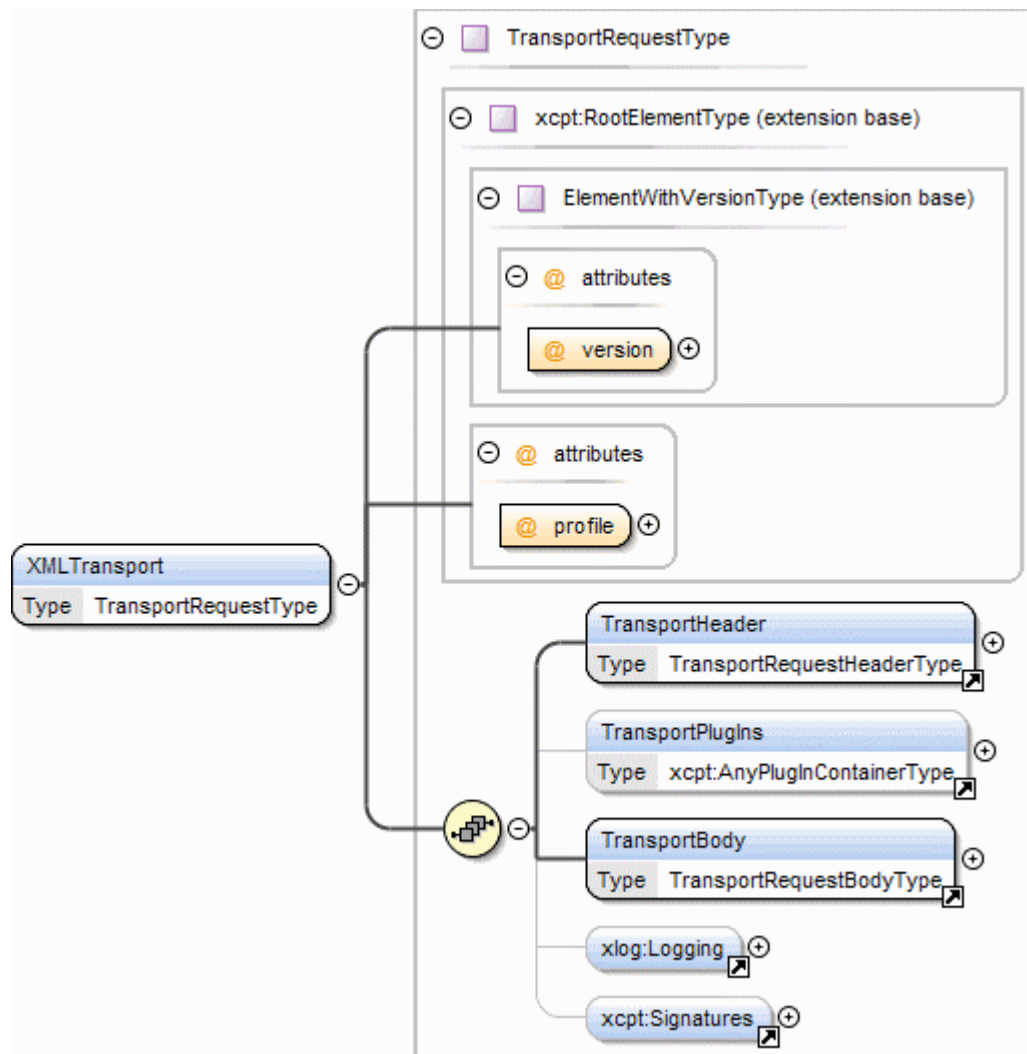
<http://www.extra-standard.de/namespace/request/1>

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

## Element XMLTransport

**xreq:XMLTransport**

|             |  |
|-------------|--|
| Inhalt      | Enthält eine Instanz der Transportebene eines eXtra Requests |
| Datentyp    | ##elements   |
| Benutzt von | -  |



### Attribute

|          | <b>@profile</b>                                   | <i>erforderlich</i> |
|----------|---|---------------------|
| Inhalt   | Enthält die ID des eXtra Profils                  |                     |
| Datentyp | xsd:anyURI  |                     |
|          | <b>@version</b>                                   | <i>erforderlich</i> |
| 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:XMLTransport ist das Wurzelement 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.

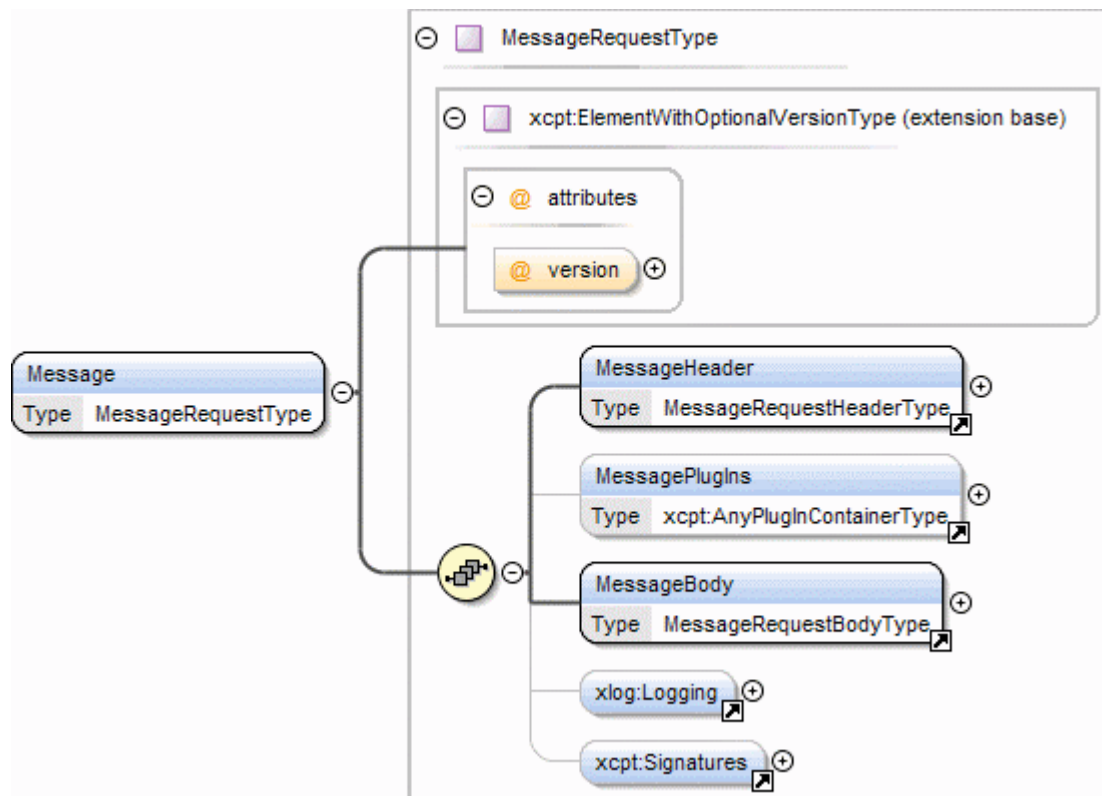
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<xreq:XMLTransport
  xmlns:xreq="http://www.extra-standard.de/namespace/request/1"
  xmlns:xcpt="http://www.extra-standard.de/namespace/components/1"
  version="http://www.extra-standard.de/version/1"
  profile="http://www.myCompany.de/profile/myProfile/1">
  <xreq:TransportHeader>
    <xcpt:Sender>
      <xcpt:SenderID></xcpt:SenderID>
    </xcpt:Sender>
    <xcpt:Receiver>
      <xcpt:ReceiverID></xcpt:ReceiverID>
    </xcpt:Receiver>
    <xcpt:RequestDetails>
      <xcpt:RequestID>111111</xcpt:RequestID>
    </xcpt:RequestDetails>
  </xreq:TransportHeader>
  <xreq:TransportBody>
    <xcpt:Data>
      <xcpt:AnyXML></xcpt:AnyXML>
    </xcpt:Data>
  </xreq:TransportBody>
</xreq:XMLTransport>
```

## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXtra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| 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`.

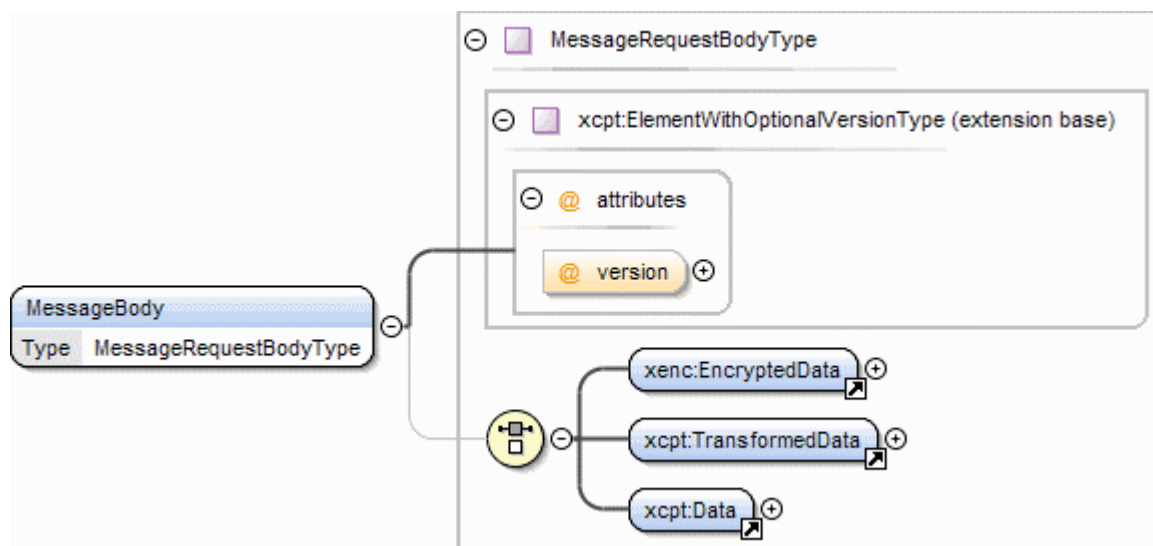
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| Datentyp | xsd:anyURI   |

## Profilierung

| Komponente           | REQ | OPT | FOR      | MUL | IGN | VAL |
|----------------------|-----|-----|----------|-----|-----|-----|
| xenc:EncryptedData   | +   | +   | +        |     |     |     |
| xcpt:TransformedData | +   | +   | +        |     |     |     |
| xcpt:Data            |     |     | IMPLICIT |     |     |     |

## Beschreibung

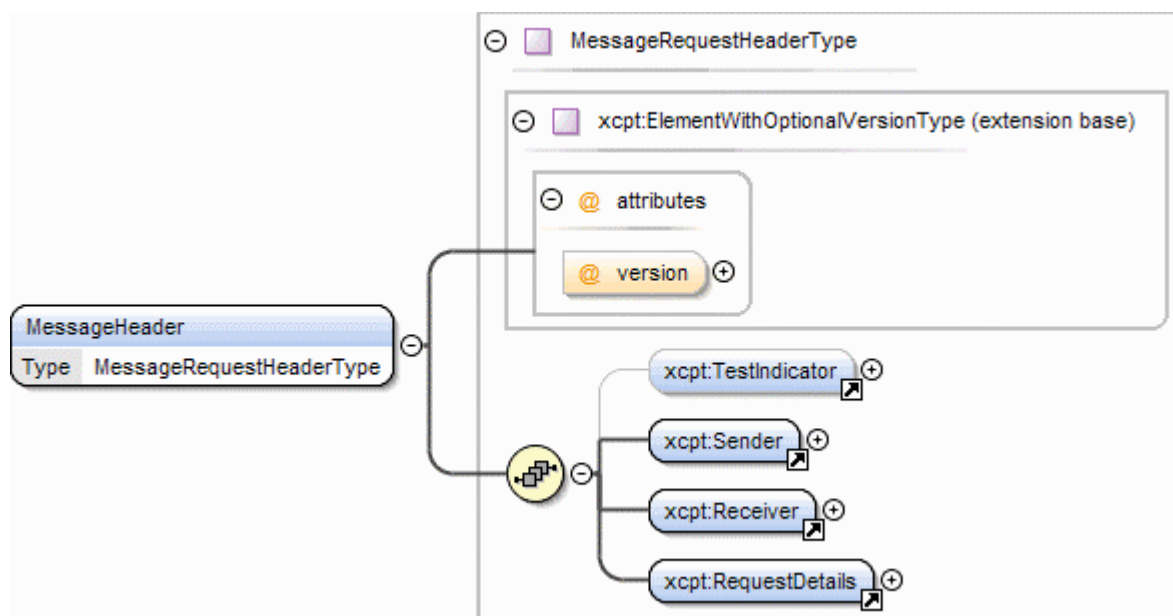
xreq:MessageBody enthält den Body einer Instanz der Nachrichtenebene eines eXTra Requests. Im Body sind die Nutzdaten in verschlüsselter oder unverschlüsselter Form abgelegt.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element MessageHeader

|             | <b>xreq:MessageHeader</b>  |
|-------------|--|
| Inhalt      | Enthält den Header einer Instanz der Nachrichtenebene eines eXtra Requests |
| Datentyp    | ##elements   |
| Benutzt von | xreq:Message   |



### Attribute

|          | <b>@version</b>  | <i>optional</i> |
|----------|--|-----------------|
| Inhalt   | Enthält die Versionsbezeichnung des eXtra Schemas. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXtra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |                 |
| 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.

### Beispiel

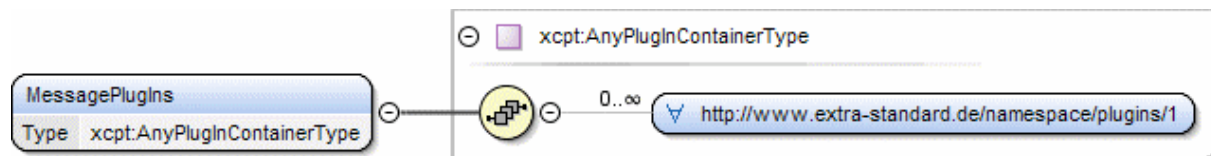
```
<?xml version="1.0" encoding="UTF-8"?>
```



## 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.

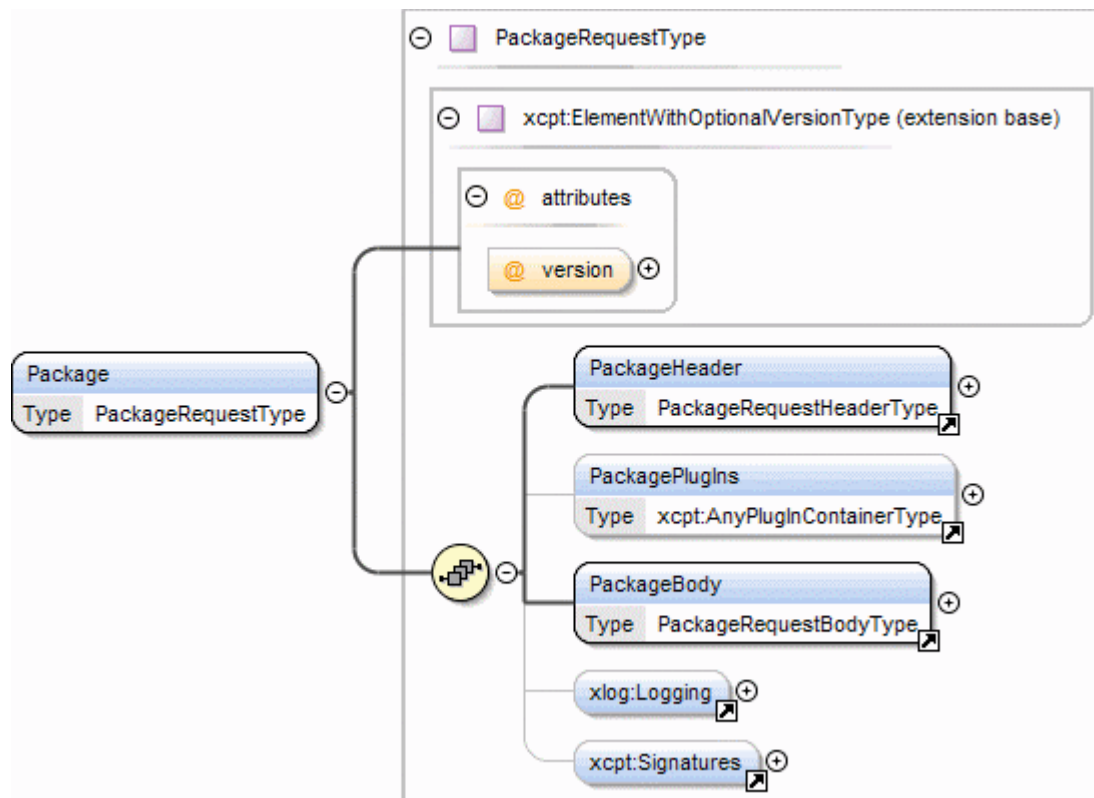
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXtra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| 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.

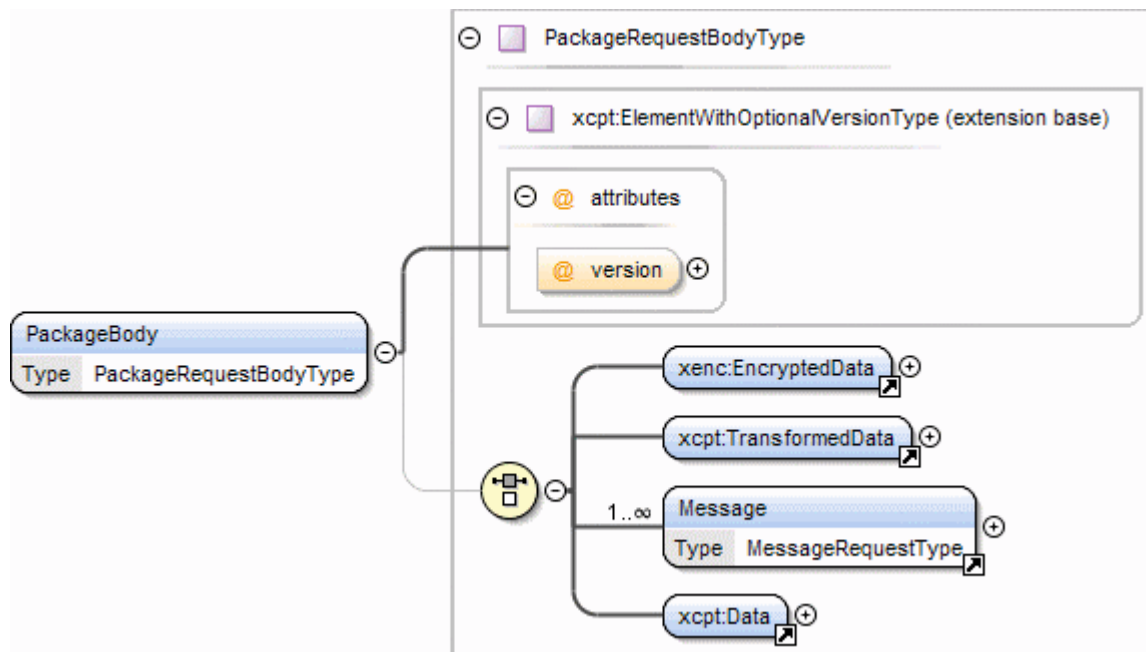
## Beispiel

```
<?xml version="1.0" encoding="UTF-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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXtra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| Datentyp | xsd:anyURI   |

### Profilierung

| Komponente           | REQ | OPT | FOR      | MUL | IGN | VAL |
|----------------------|-----|-----|----------|-----|-----|-----|
| xenc:EncryptedData   | +   | +   | +        |     |     |     |
| xcpt:TransformedData | +   | +   | +        |     |     |     |
| xcpt:Data            |     |     | IMPLICIT |     |     |     |

### Beschreibung

`xreq:PackageBody` 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`).



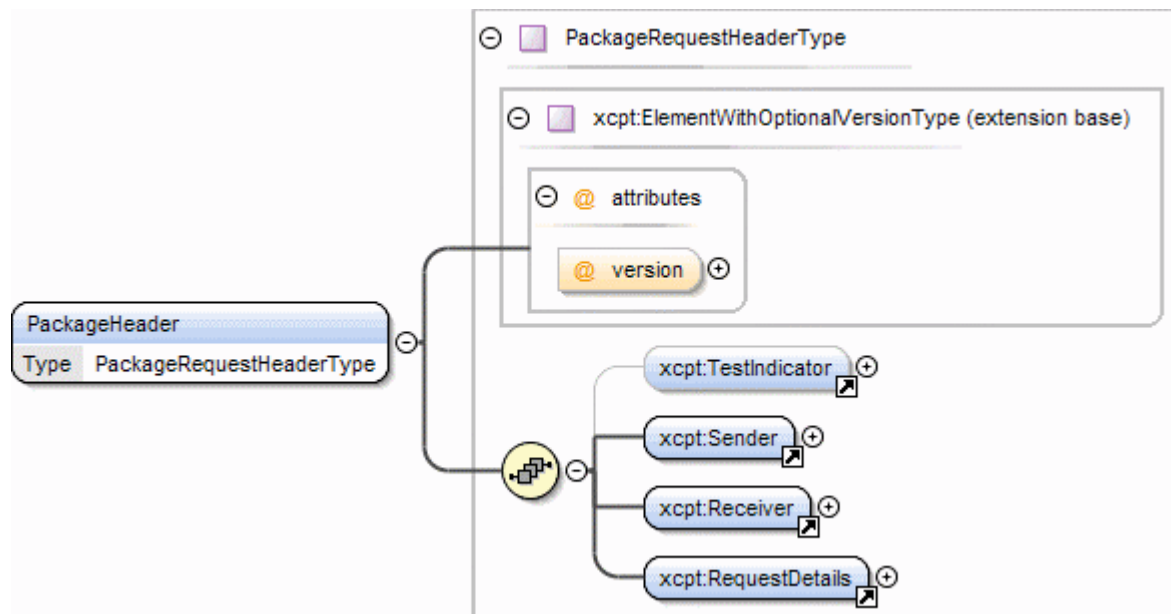
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXtra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| Datentyp | xsd:anyURI   |

### Profilierung

Der Inhalt des Elementes unterliegt nicht der Profilierung.

### Beschreibung

`xreq:PackageHeader` den Header einer Instanz der Paketebene eines eXtra Requests.

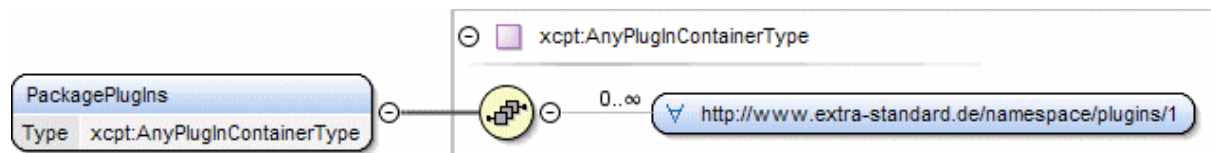
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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.

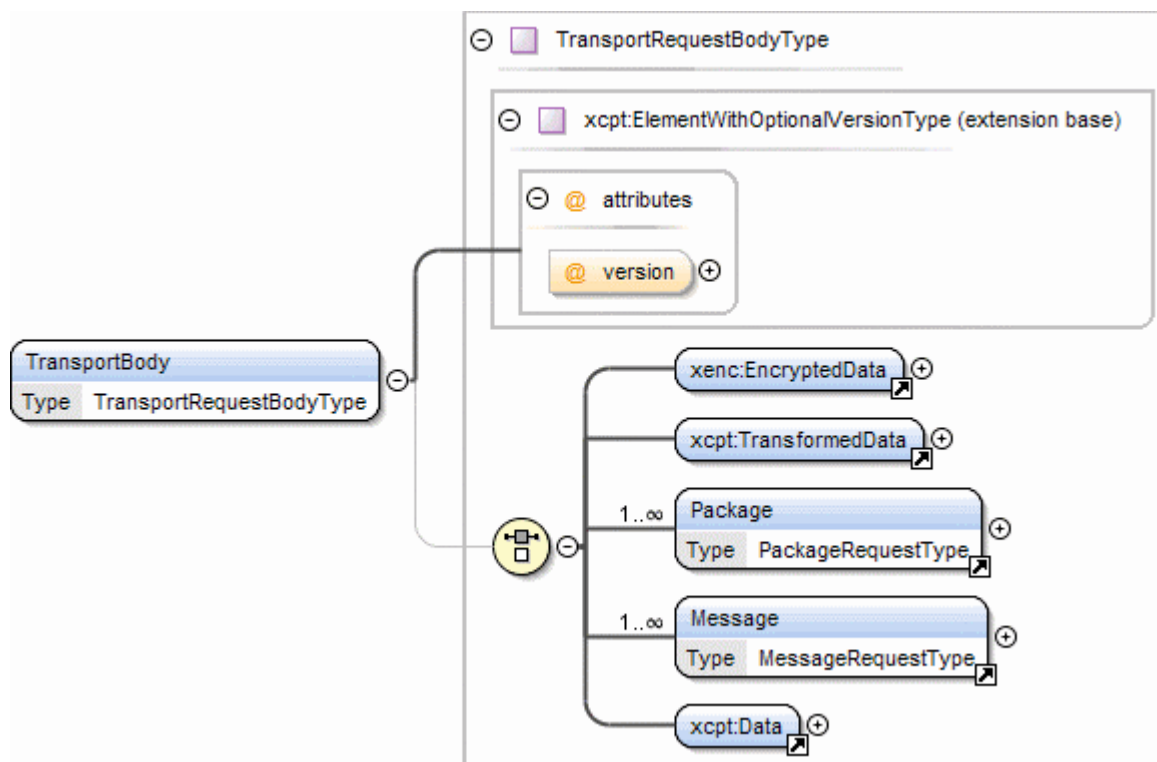
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| 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].

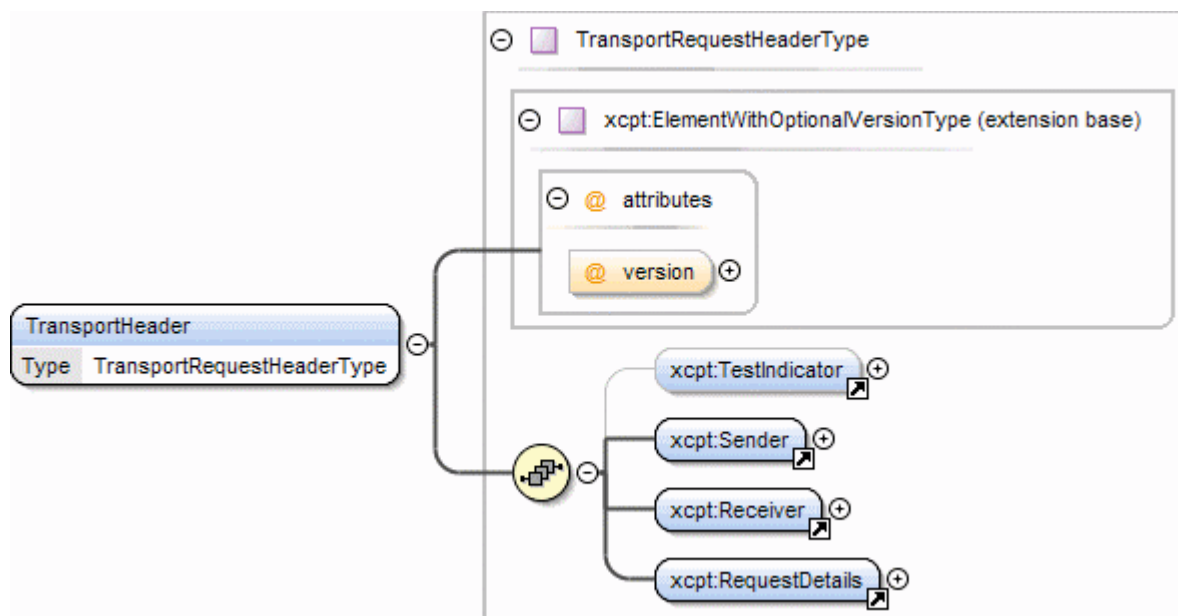
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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.

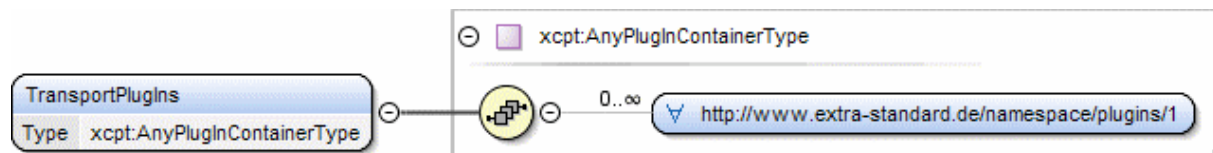
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Nachrichtentyp Response [normativ]

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

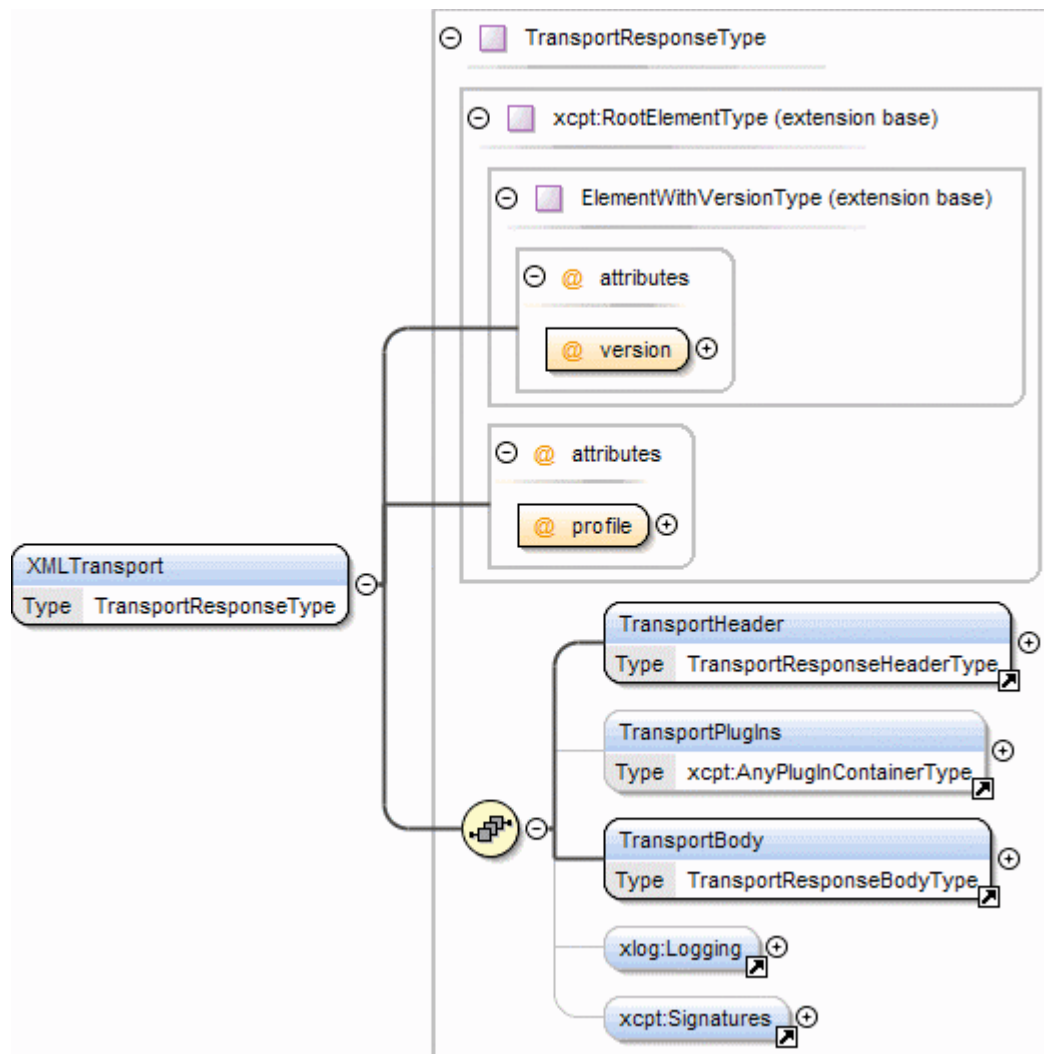
<http://www.extra-standard.de/namespace/response/1>

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

## Element XMLTransport

### xres:XMLTransport

|             |  |
|-------------|--|
| 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

| <i>Komponente</i>            | <i>REQ</i> | <i>OPT</i> | <i>FOR</i> | <i>MUL</i> | <i>IGN</i> | <i>VAL</i> |
|------------------------------|------------|------------|------------|------------|------------|------------|
| <b>@profile</b>              |            |            |            |            |            | +          |
| <b>xcpt:Signatures</b>       | +          | +          | +          |            | +          |            |
| <b>xlog:Logging</b>          | +          | +          | +          |            | +          |            |
| <b>xres:TransportPlugins</b> | +          | +          | +          |            | +          |            |

## Beschreibung

`xres:XMLTransport` ist das Wurzelement 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.

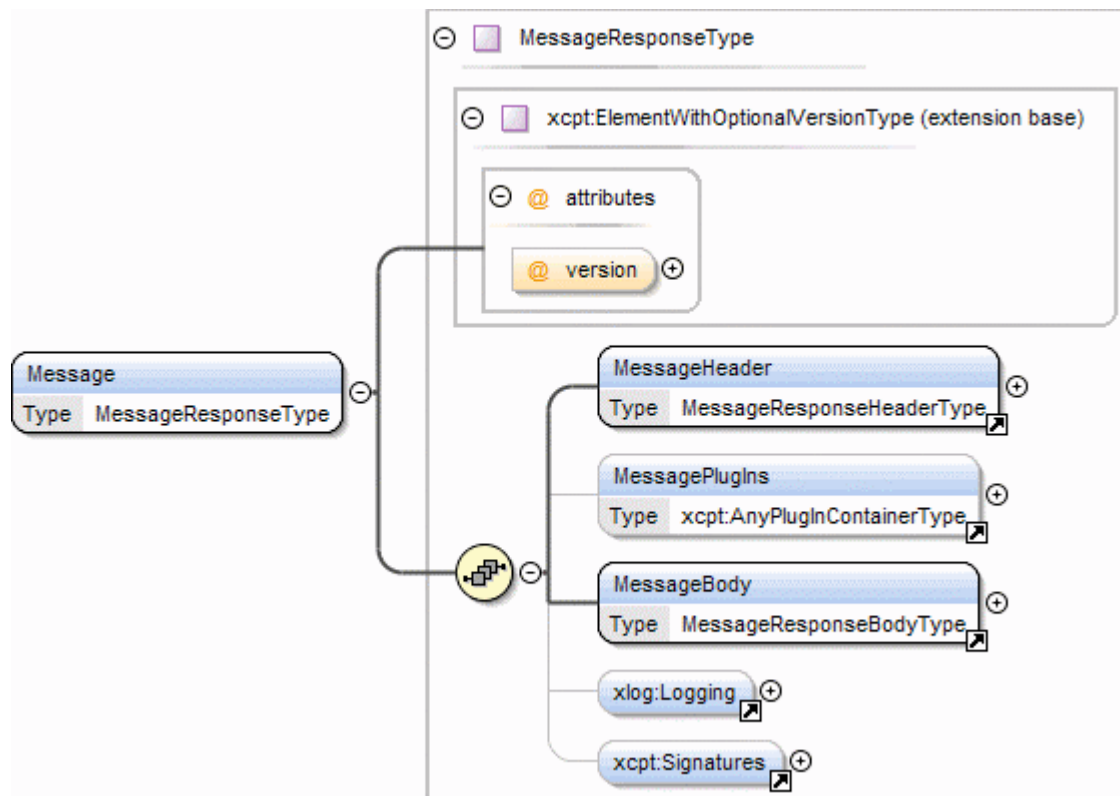
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXtra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| 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`.

### Beispiel

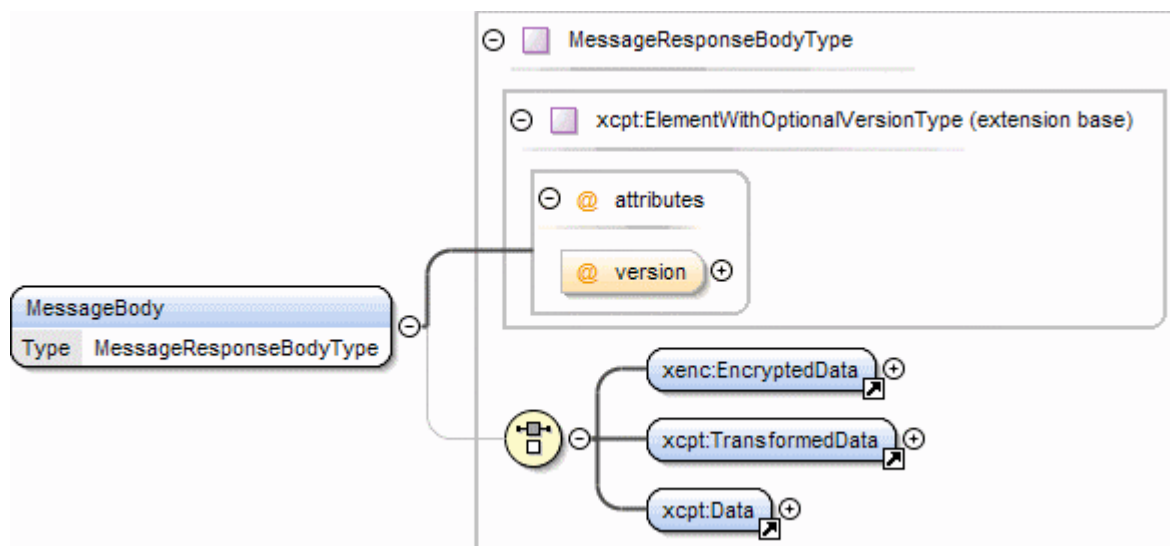
```
<?xml version="1.0" encoding="UTF-8"?>
```



## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| 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.

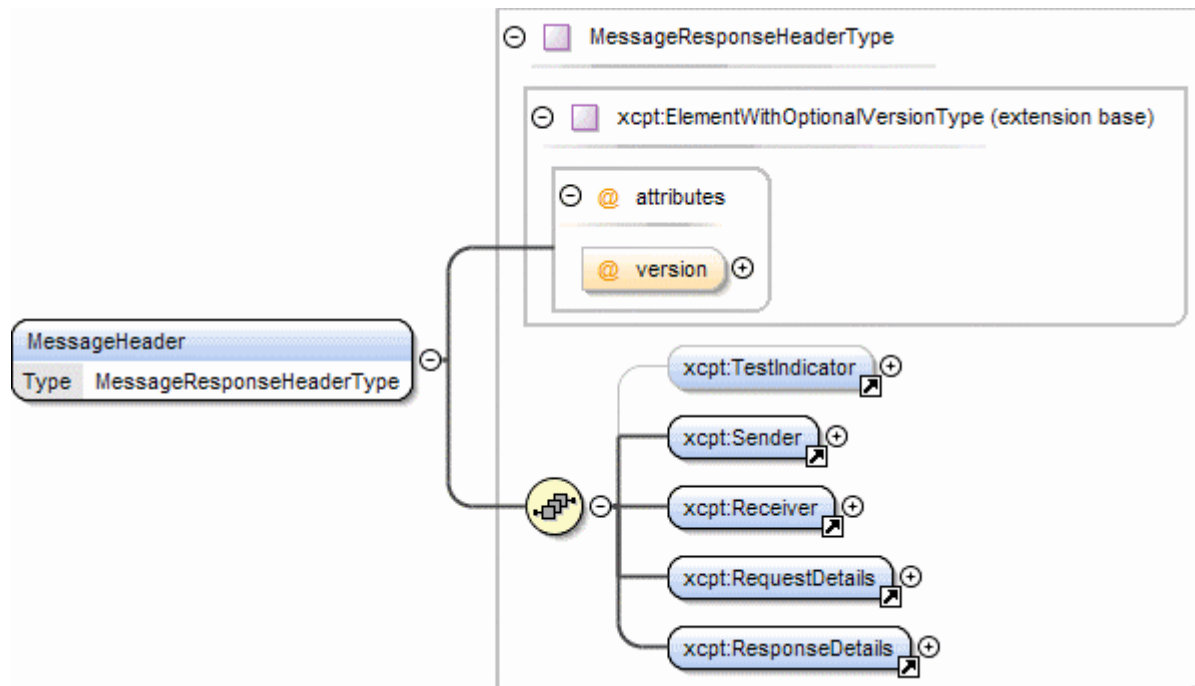
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| 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.

### Beispiel



**eXtra Transport V1.2**  
**Schnittstellenbeschreibung V1.2.1**

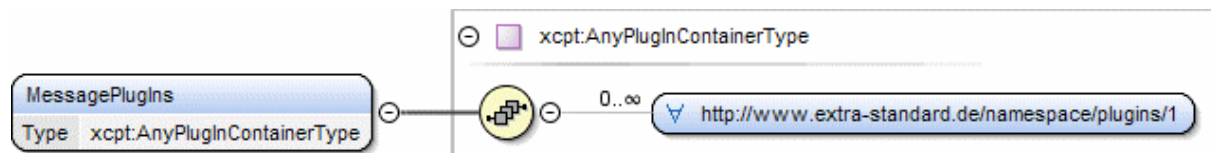
Seite: 51  
Status: **FINAL**  
Stand: **08.09.2010**

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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.

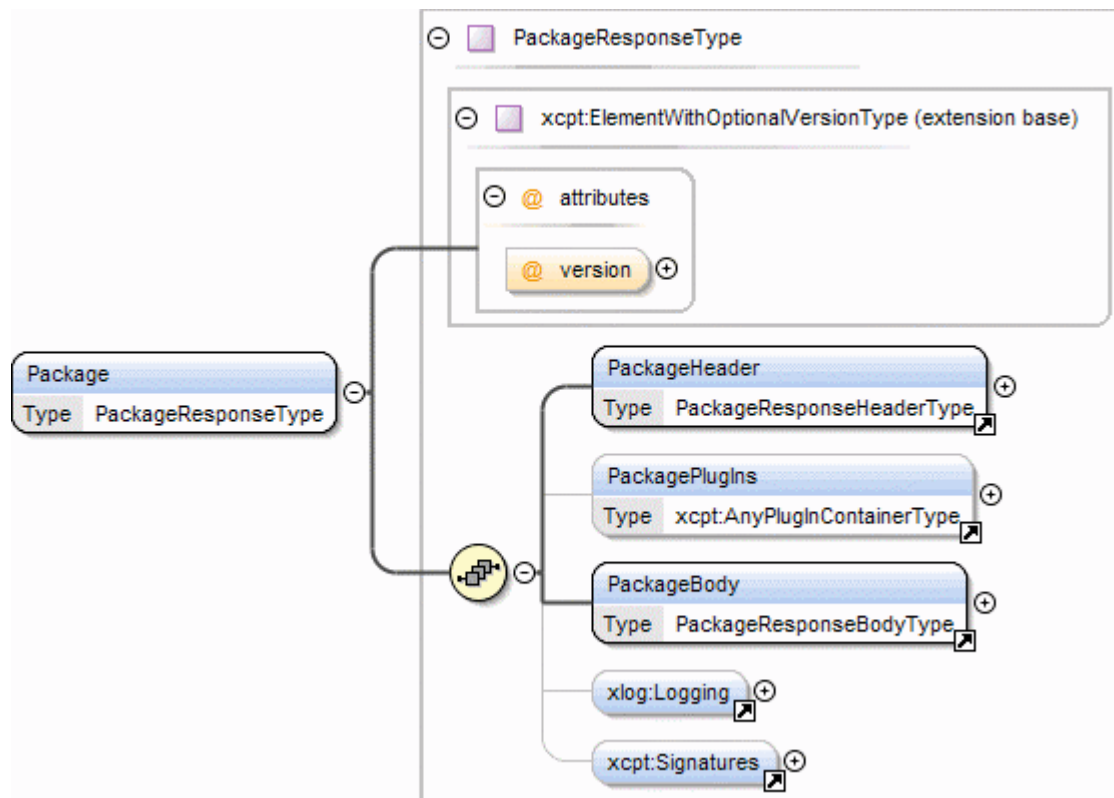
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXtra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| 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.

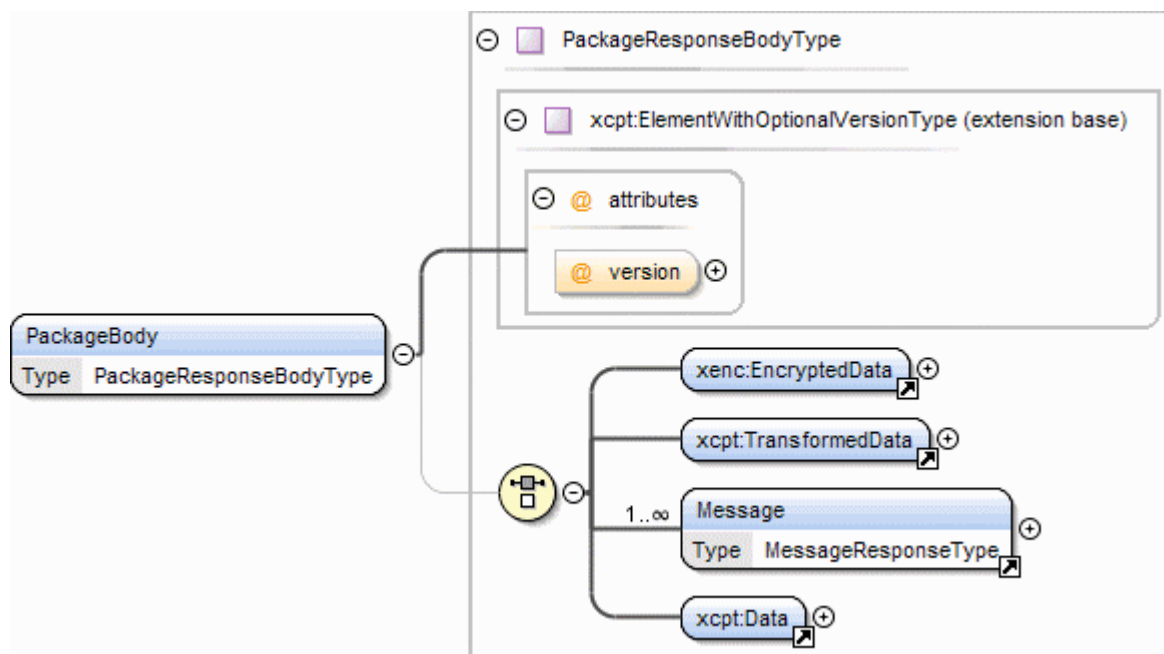
## Beispiel

```
<?xml version="1.0" encoding="UTF-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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXtra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| Datentyp | xsd:anyURI   |

### Profilierung

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

### Beschreibung

**xres:PackageBody** 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**).



### **Beispiel**

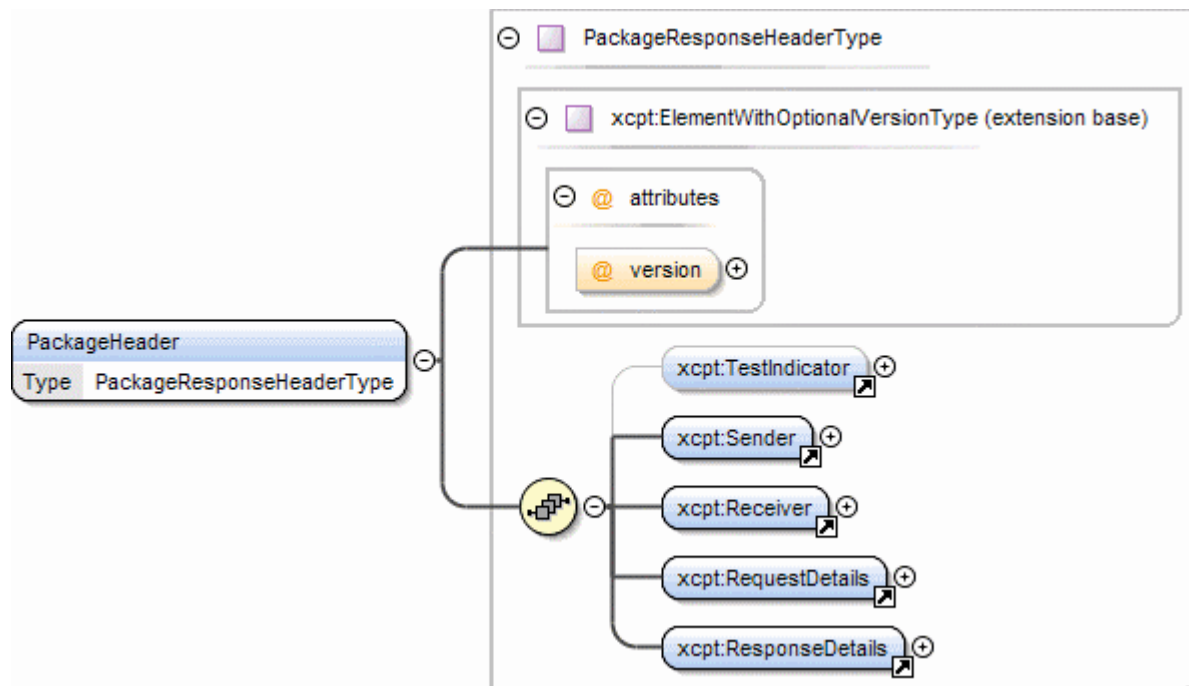
```
<?xml version="1.0" encoding="UTF-8"?>
```



## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXtra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
|--------|--|

|          |            |
|----------|------------|
| Datentyp | xsd:anyURI |
|----------|------------|

### Profilierung

Der Inhalt des Elementes unterliegt nicht der Profilierung.

### Beschreibung

`xres:PackageHeader` den Header einer Instanz der Paketebene eines eXtra Response.

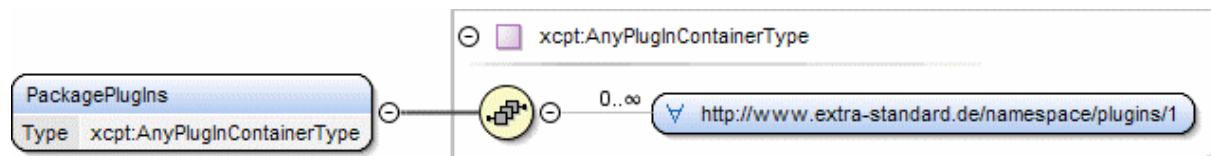
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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.

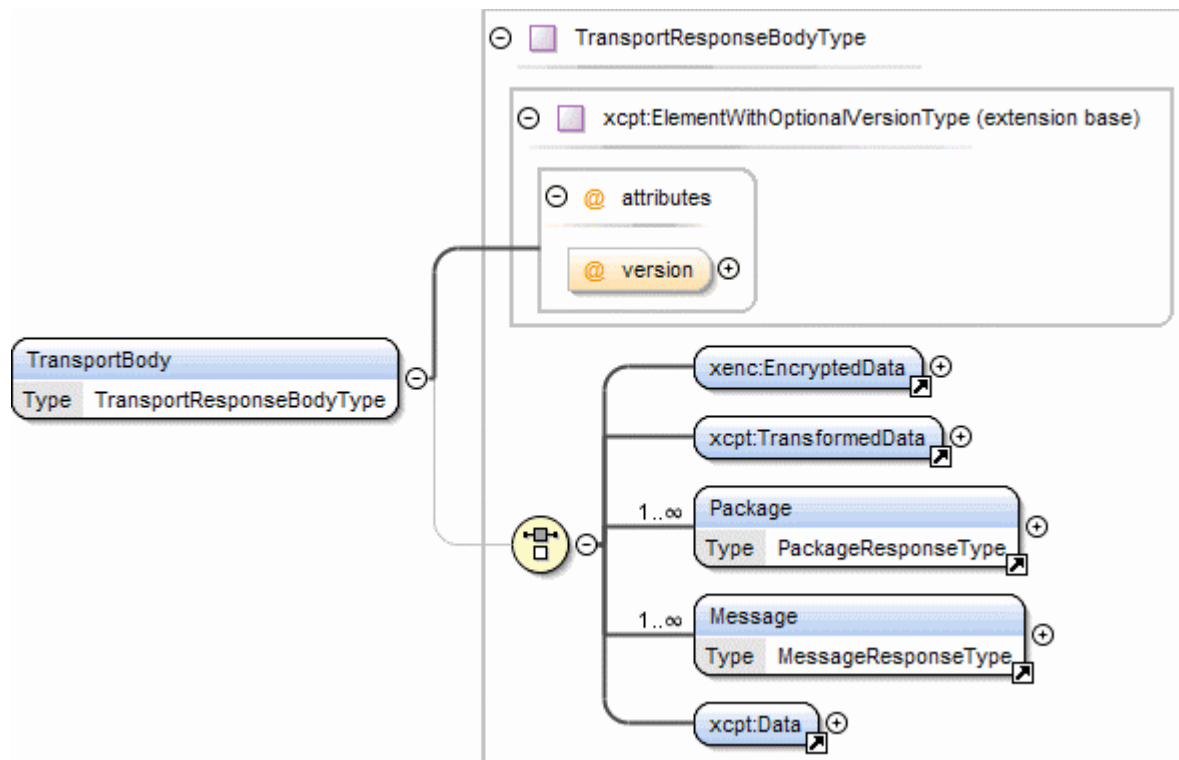
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| 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].

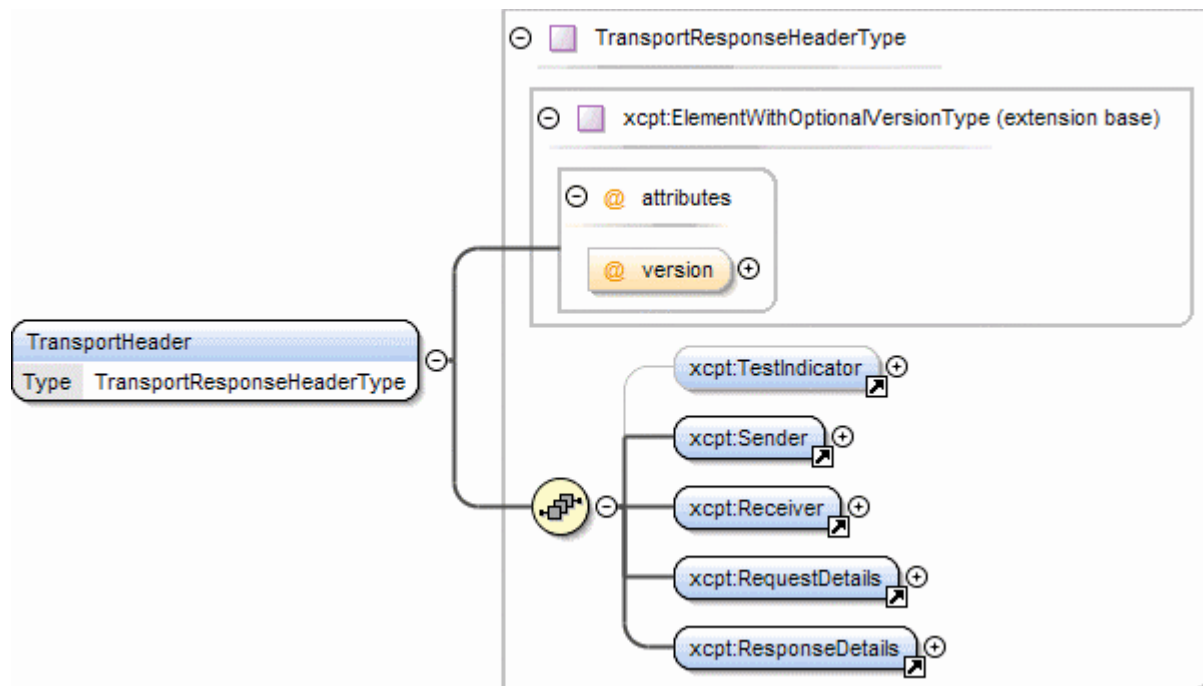
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXtra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
|--------|--|

|          |            |
|----------|------------|
| Datentyp | xsd:anyURI |
|----------|------------|

### Profilierung

Der Inhalt des Elementes unterliegt nicht der Profilierung.

### Beschreibung

`xres:TransportHeader` enthält den Header der Transportebene eines eXtra Requests.

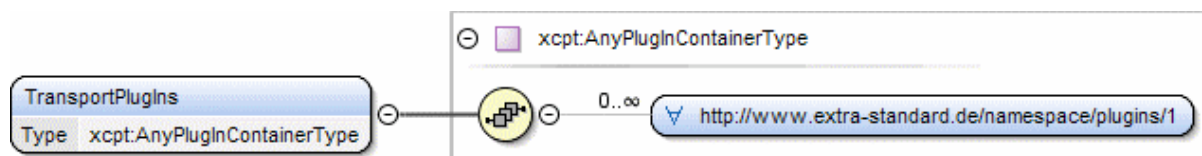
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## **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

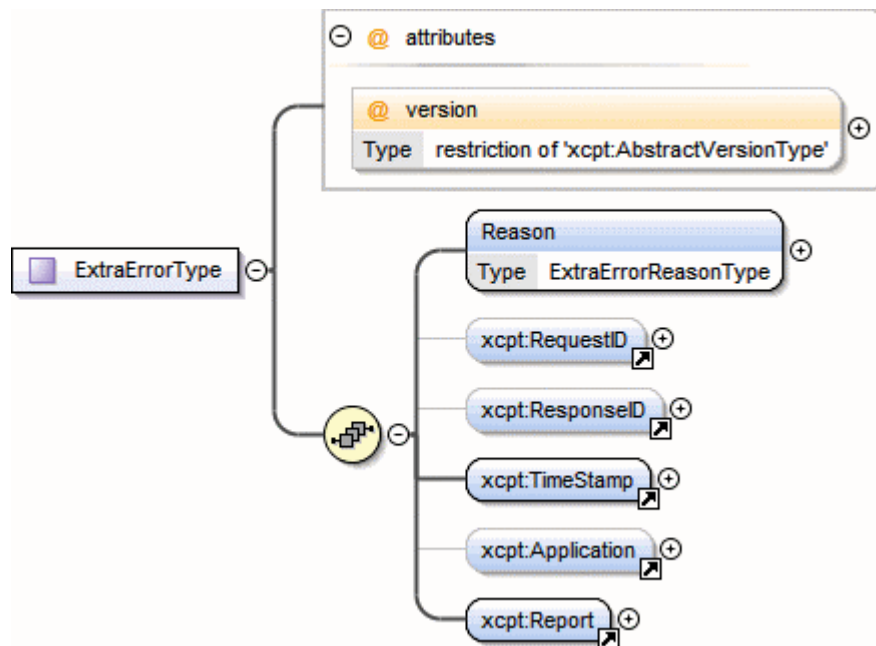
<http://www.extra-standard.de/namespace/service/1>

Der lokale Name des Wurzelementes ist `ExtraError`. Es wird empfohlen, den Namensraumpräfix `xsrv` zu verwenden.

## 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.

## Beispiel

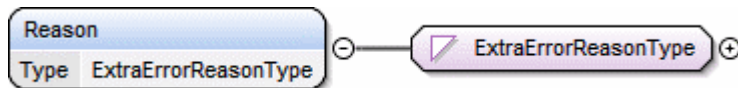
```
<?xml version="1.0" encoding="UTF-8"?>
```



## 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_TEMPORARILY_UNAVAILABLE)

[http://www.extra-standard.de/reason/SERVICE\\_PERMANENTLY\\_UNAVAILABLE](http://www.extra-standard.de/reason/SERVICE_PERMANENTLY_UNAVAILABLE)

[http://www.extra-standard.de/reason/SERVICE\\_FAILURE](http://www.extra-standard.de/reason/SERVICE_FAILURE)

[http://www.extra-standard.de/reason/INVALID\\_REQUEST](http://www.extra-standard.de/reason/INVALID_REQUEST)

<http://www.extra-standard.de/reason/UNSPECIFIED>

## Beispiel

```
<?xml version="1.0" encoding="UTF-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 Wurzelement. 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:

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

## Element Value

### xplg:Value

|                    |  |
|--------------------|--|
| <i>Inhalt</i>      | Enthält den Schlüssel eines Zertifikates in Base64 Kodierung |
| <i>Datentyp</i>    | xs:base64Binary  |
| <i>Benutzt von</i> | xplg:X509EncCertificate, xplg:X509ReplyCertificate           |



## Beschreibung

Enthält den Schlüssel eines Zertifikates in Base64 Kodierung.

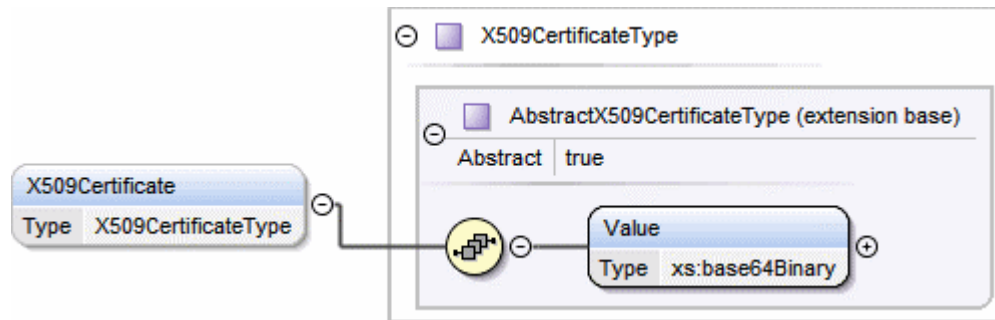
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element X509Certificate

### xplg:X509Certificate

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



## Beschreibung

Beschreibt ein X509 Zertifikat.

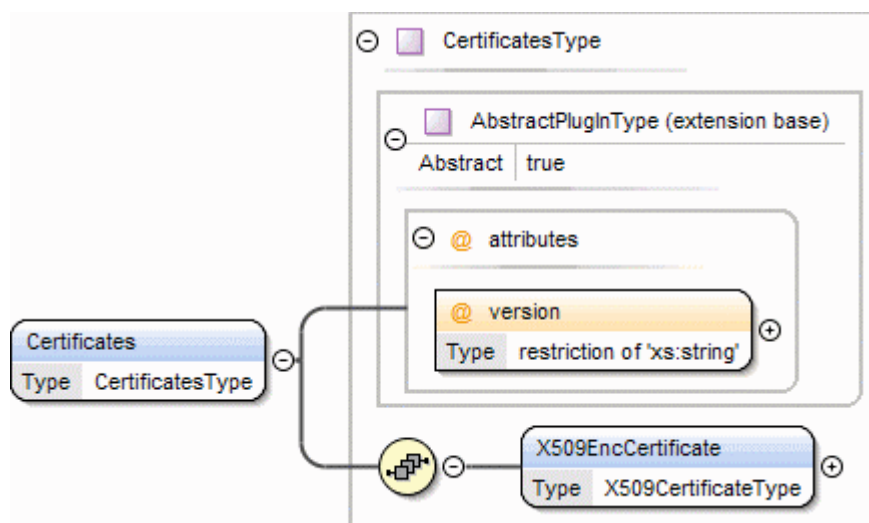
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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,<br>xreq:TransportPlugIns, xres:MessagePlugIns,<br>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.

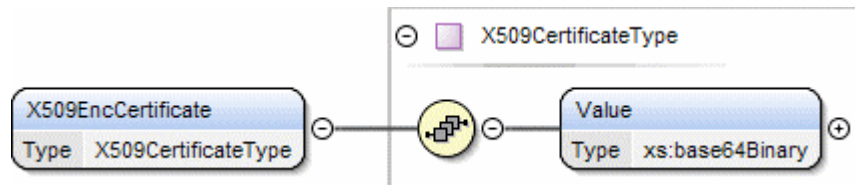
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element X509EncCertificate

**xplg:X509EncCertificate**

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



### Beschreibung

Beschreibt ein X509 Zertifikat für Verschlüsselungszwecke.

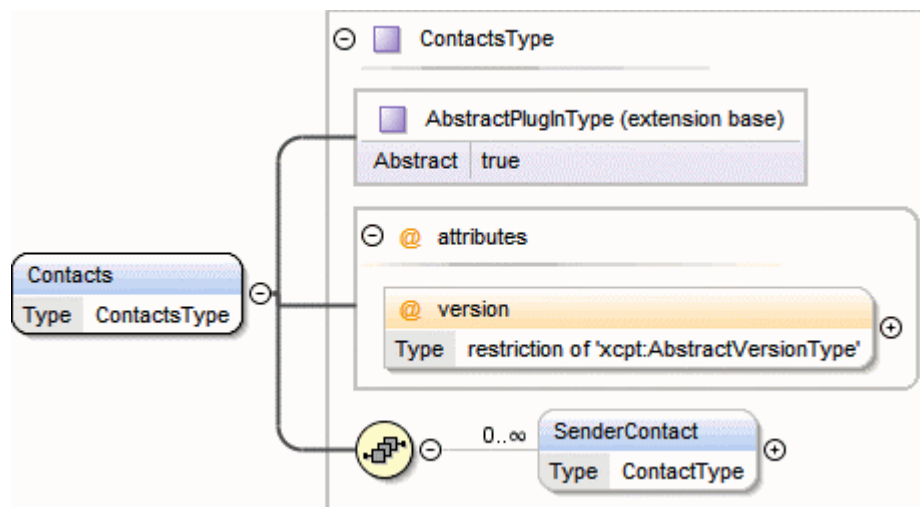
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Plug-In Contacts

### xplg:Contacts

|             |  |
|-------------|--|
| Inhalt      | Beschreibt X509 Zertifikate für die Verschlüsselung von Nachrichten.   |
| Datentyp    | ##elements   |
| Benutzt von | xreq:MessagePlugIns, xreq:PackagePlugIns,<br>xreq:TransportPlugIns, xres:MessagePlugIns,<br>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.

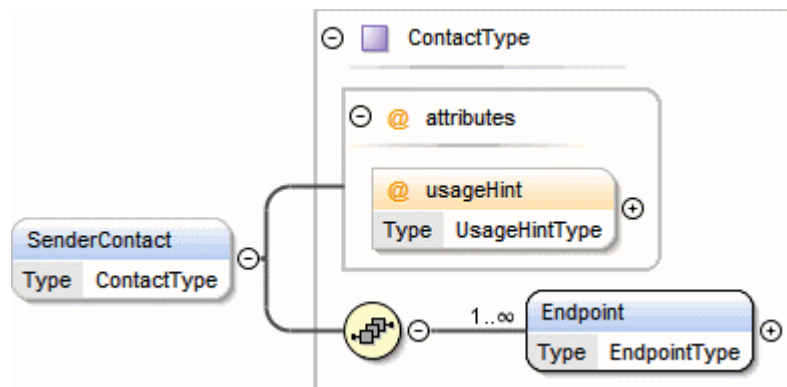
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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.

### Beispiel

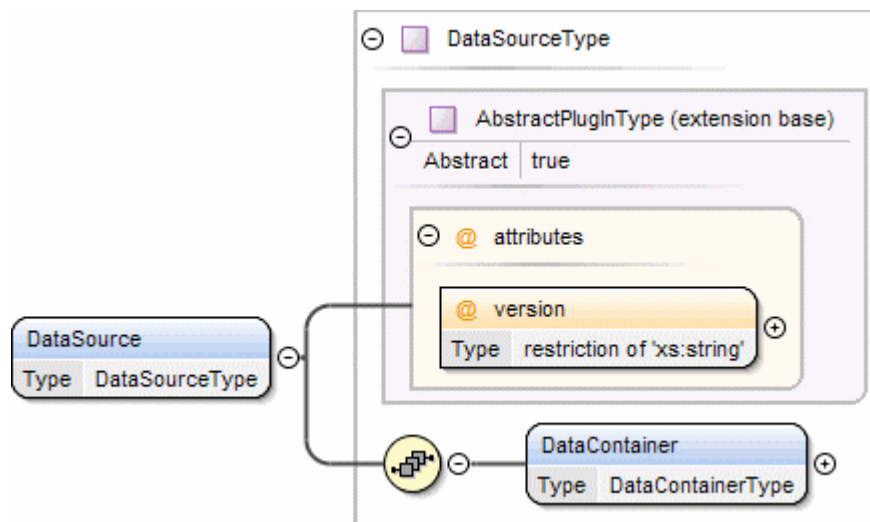
```
<?xml version="1.0" encoding="UTF-8"?>
```



## Plug-In DataSource

### xplg:DataSource

|             |  |
|-------------|--|
| Inhalt      | Beschreibt in abstrakter Form die Quelle, aus der Nutzdaten stammen.   |
| Datentyp    | ##elements   |
| Benutzt von | xreq:MessagePlugIns, xreq:PackagePlugIns,<br>xreq:TransportPlugIns, xres:MessagePlugIns,<br>xres:PackagePlugIns, xres:TransportPlugIns |



## Attribute

### @version

*optional*

|          |  |
|----------|--|
| Inhalt   | Enthält die Versionsbezeichnung des Plug-Ins, derzeit 1.0. |
| Datentyp | xsd:string   |

## Beschreibung

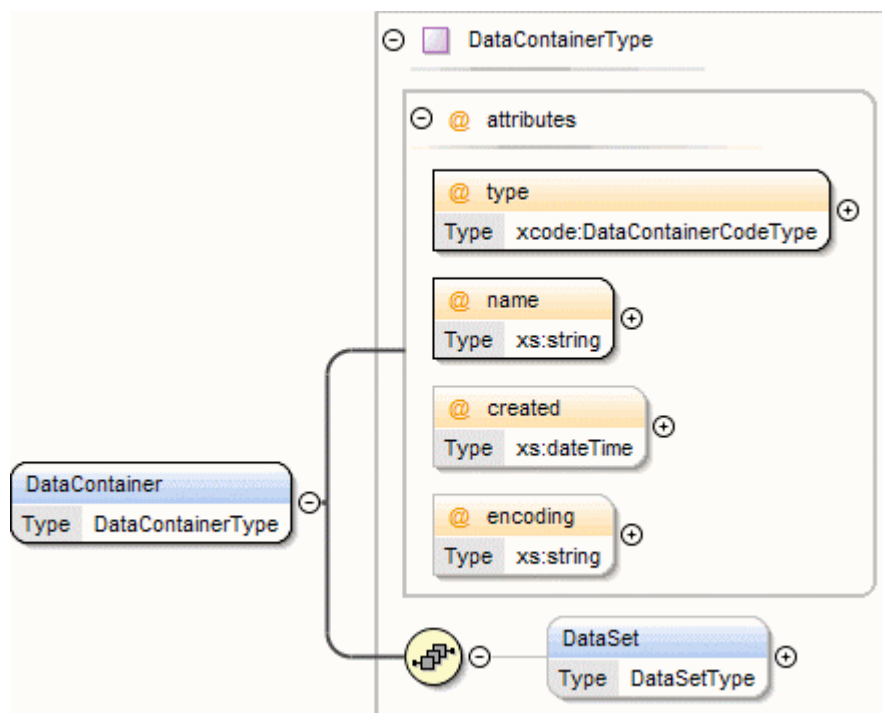
Beschreibt in abstrakter Form die Quelle, aus der Nutzdaten stammen. Es kann sich dabei z.B. um eine Datei oder eine Datenbank handeln.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element DataContainer

|             |   |
|-------------|---|
|             | <b>xplg:DataContainer</b>   |
| 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 `DataSet` ermöglicht die Beschreibung der Nutzdaten.

### Attribute

|          |   |                     |
|----------|---|---------------------|
|          | <b>@type</b>                                | <i>erforderlich</i> |
| Inhalt   | Typ des Containers                          |                     |
| Datentyp | Codewerte <a href="#">DataContainerCode</a> |                     |
|          | <b>@name</b>                                | <i>erforderlich</i> |
| Inhalt   | Bezeichnung des Containers                  |                     |
| Datentyp | xsd:string                                  |                     |
|          | <b>@created</b>                             | <i>optional</i>     |
| Inhalt   | Datum, an dem der Container erzeugt wurde.  |                     |

Datentyp      xsd:dateTime

**@encoding**

*optional*

Inhalt      Bezeichnung des Zeichensatzes

Datentyp      xsd:string

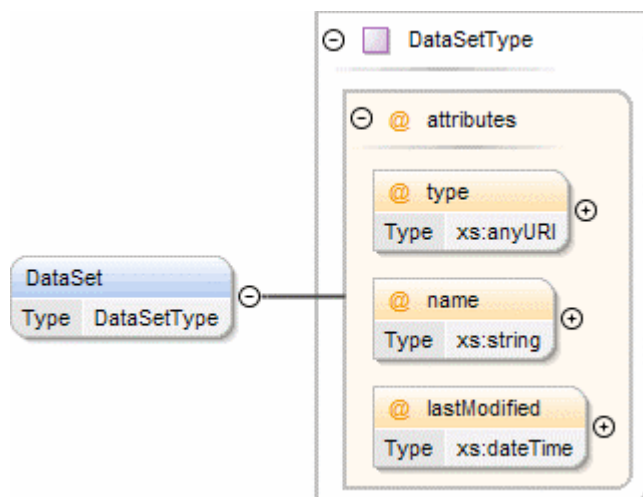
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element DataSet

### xplg:DataSet

|             |                       |
|-------------|-----------------------|
| Inhalt      | Beschreibt Nutzdaten. |
| Datentyp    | ##empty               |
| Benutzt von | xplg:DataContainer    |



## Beschreibung

Beschreibt die Nutzdaten.

## Attribute

|          | @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   |              |

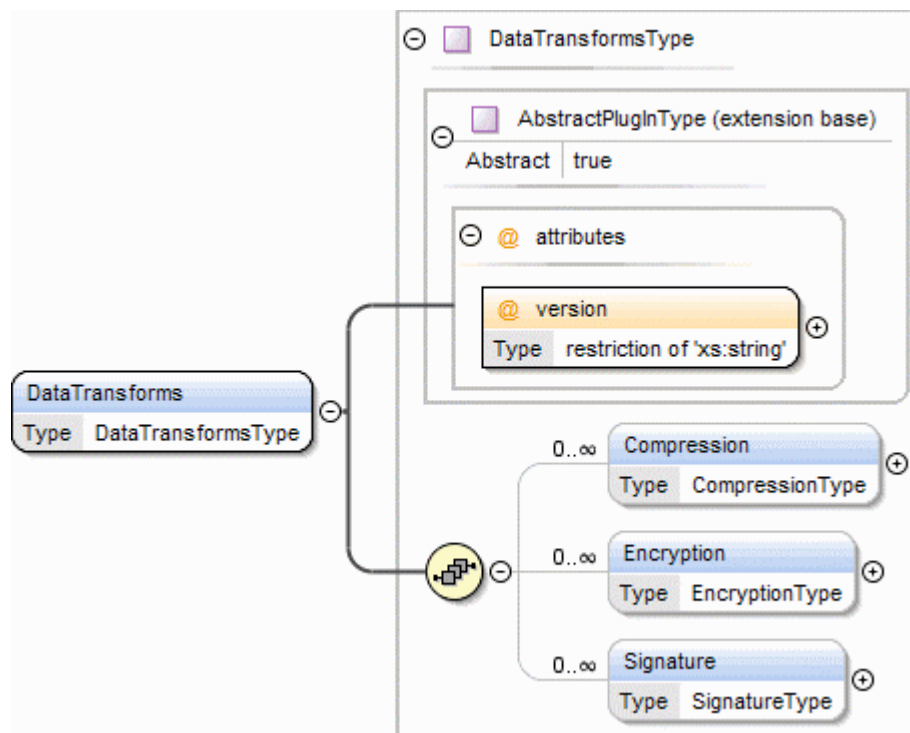
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Plug-In DataTransforms

### xplg:DataTransforms

|             |  |
|-------------|--|
| Inhalt      | Beschreibt Transformationen wie Verschlüsselung und Komprimierung.   |
| Datentyp    | ##elements   |
| Benutzt von | xreq:MessagePlugIns, xreq:PackagePlugIns,<br>xreq:TransportPlugIns, xres:MessagePlugIns,<br>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.

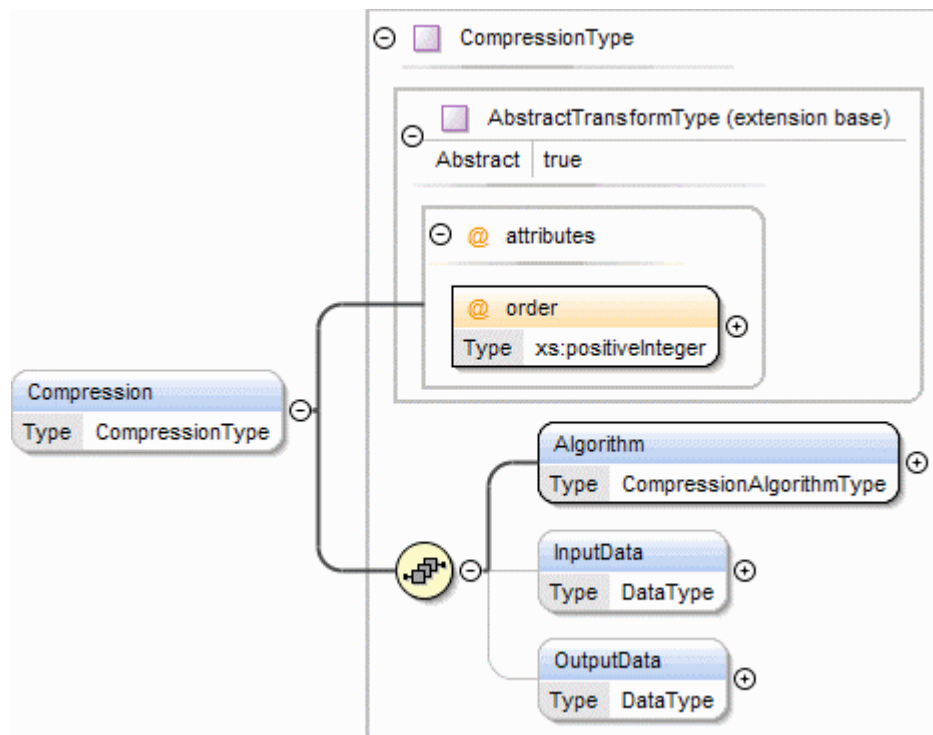
### **Beispiel**

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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                           |

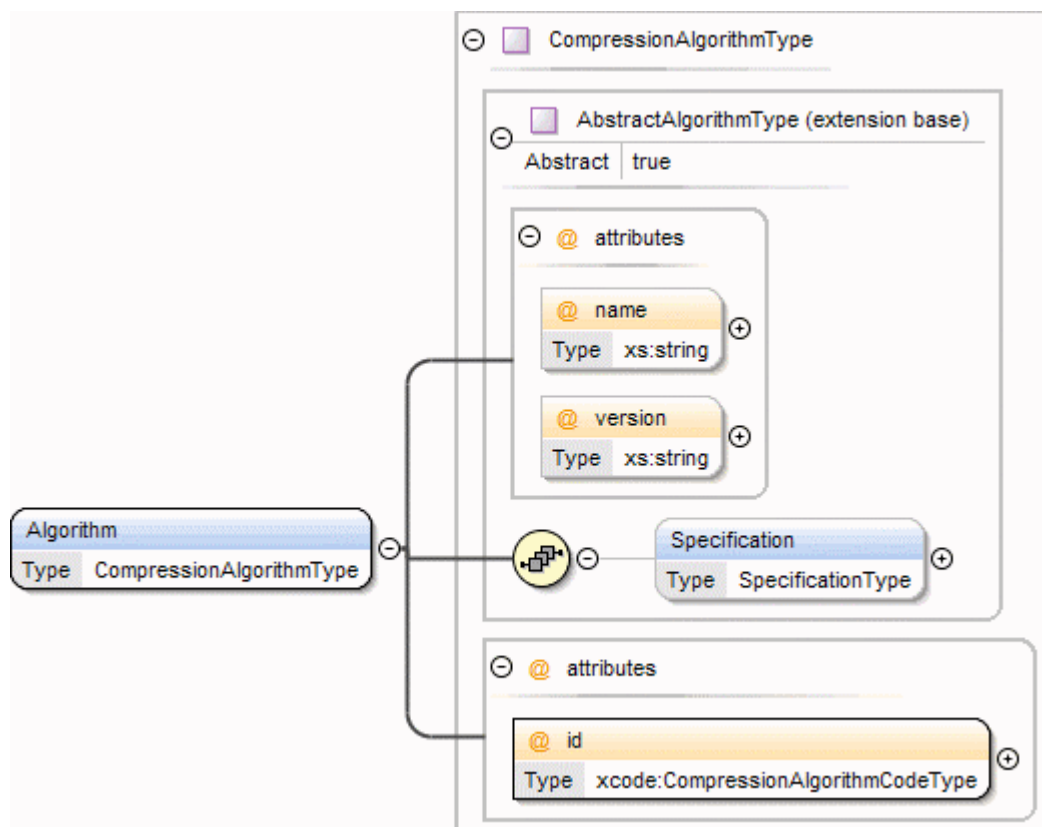
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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 <a href="#">CompressionAlgorithmCode</a> |              |
|          | @name  | Optional     |
| Inhalt   | Bezeichner des Algorithmus.                        |              |
| Datentyp | xsd:string   |              |
|          | @version   | Optional     |
| Inhalt   | Version des Algorithmus.                           |              |
| Datentyp | xsd:string   |              |



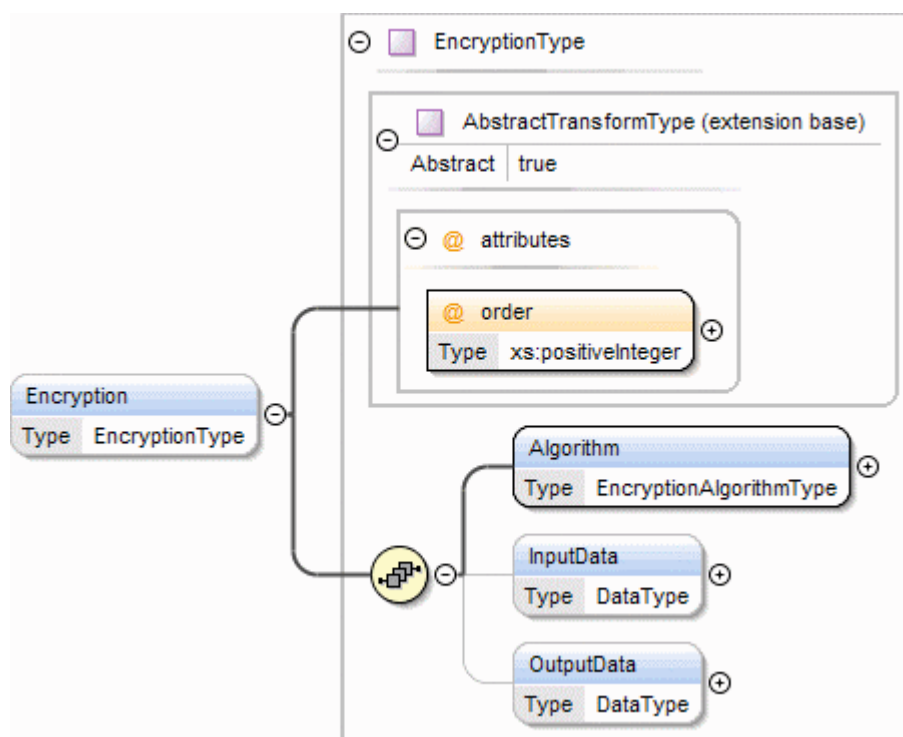
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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                           |

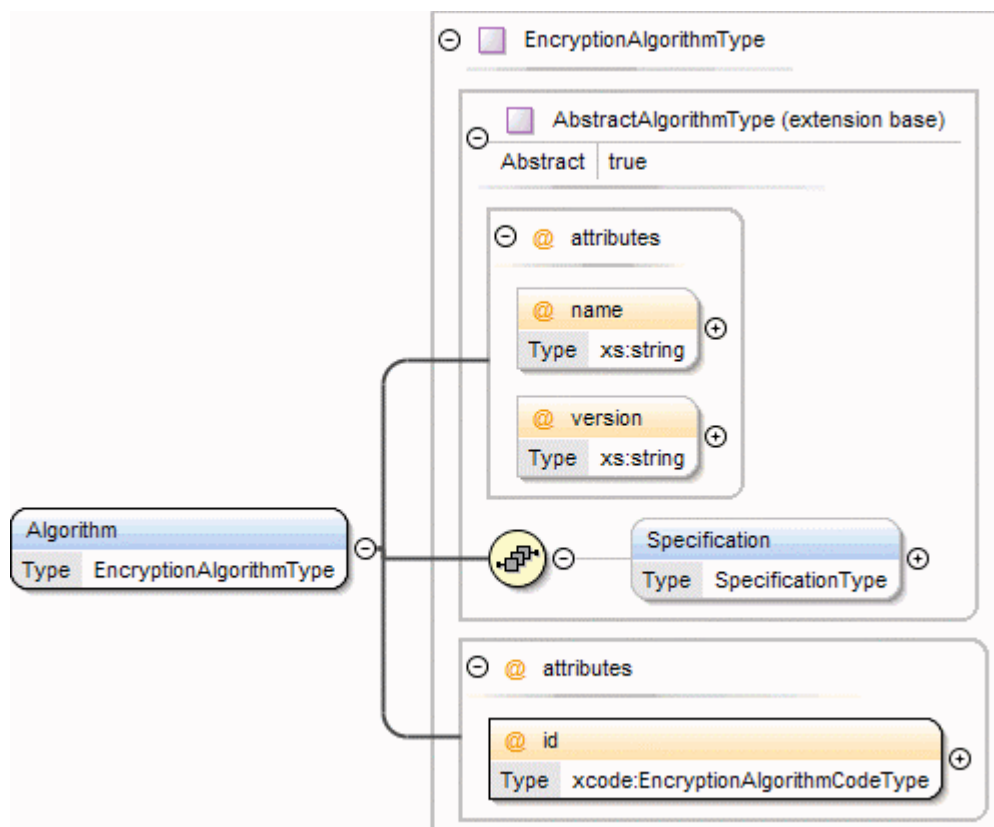
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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 <a href="#">EncryptionAlgorithmCode</a> |              |
|          | @name   | Optional     |
| Inhalt   | Bezeichner des Algorithmus.                       |              |
| Datentyp | xsd:string  |              |
|          | @version  | Optional     |
| Inhalt   | Version des Algorithmus.                          |              |
| Datentyp | xsd:string  |              |

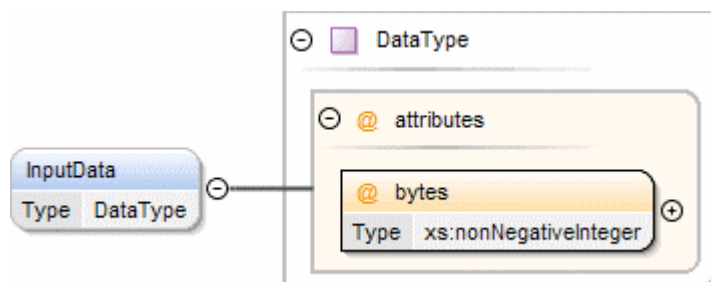
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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   |

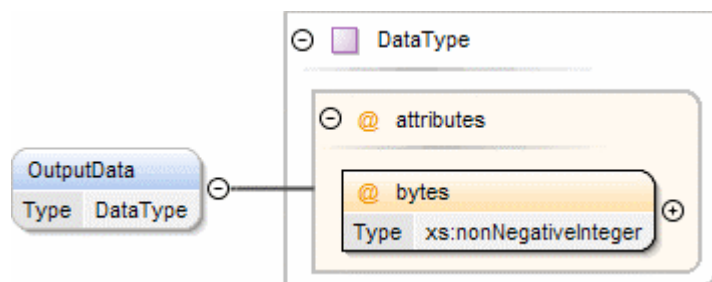
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element *OutputData*

### xplg:OutputData

|             |  |
|-------------|--|
| Inhalt      | Beschreibt den Zustand von Daten <i>nach</i> 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   |

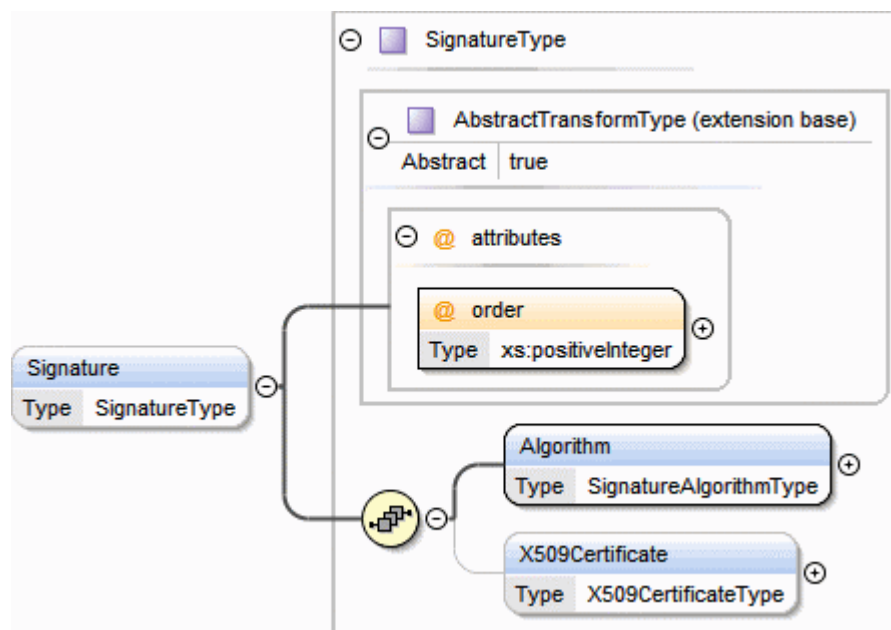
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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 beigelegt 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                                    |              |

## Beispiel

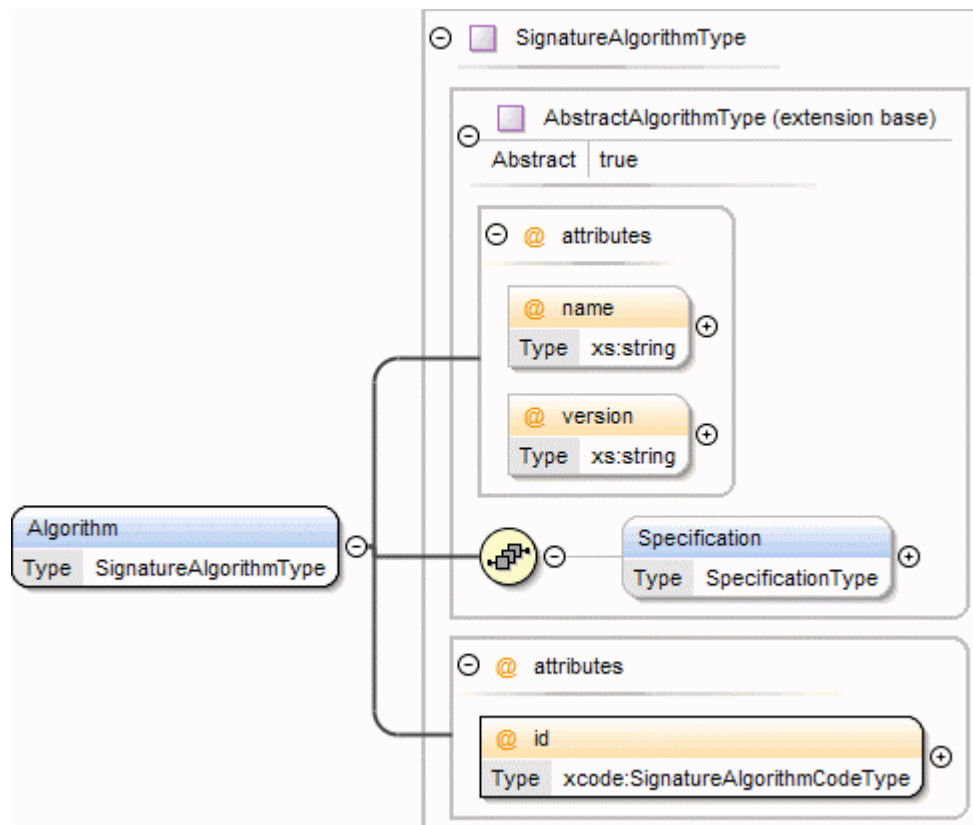
```
<?xml version="1.0" encoding="UTF-8"?>
```



## 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 <a href="#">SignatureAlgorithmCode</a> |              |
|          | @name  | Optional     |
| Inhalt   | Bezeichner des Verfahrens.                       |              |
| Datentyp | xsd:string                                       |              |
|          | @version   | Optional     |
| Inhalt   | Version des Verfahrens.                          |              |
| Datentyp | xsd:string                                       |              |

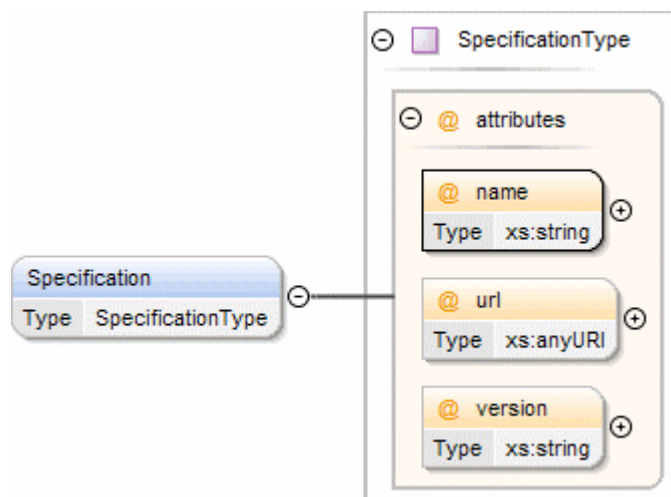
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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                         | <i>erforderlich</i> |
|----------|-------------------------------|---------------------|
| Inhalt   | Bezeichnung der Spezifikation |                     |
| Datentyp | xsd:string                    |                     |
|          | @url                          | <i>optional</i>     |
| Inhalt   | URL der Spezifikation         |                     |
| Datentyp | xsd:anyURI                    |                     |
|          | @version                      | <i>optional</i>     |
| Inhalt   | Version der Spezifikation     |                     |
| Datentyp | xsd:string                    |                     |

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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

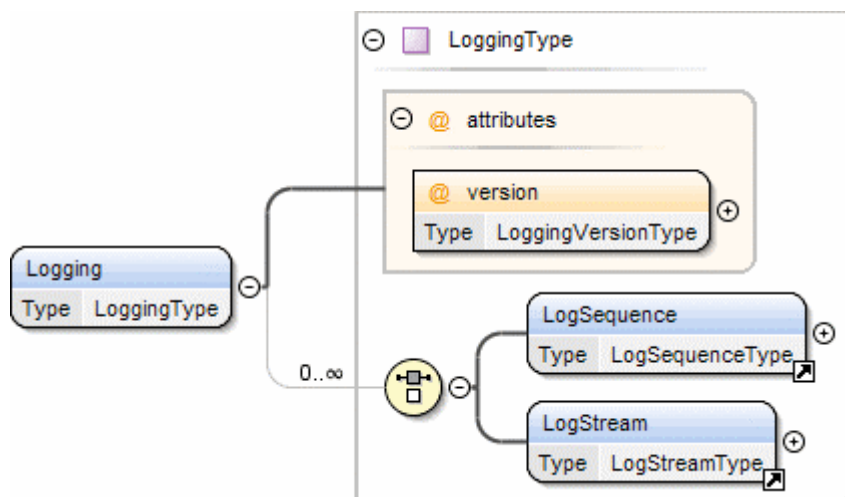
<http://www.extra-standard.de/namespace/logging/1>

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

## Element Logging

### xlog:Logging

|             |   |
|-------------|---|
| Inhalt      | Enthält das Log einer logischen Ebene des eXTra Modells   |
| Datentyp    | ##elements  |
| Benutzt von | xreq:Message, xreq:Package, xreq:XMLTransport,<br>xres:Message, xres:Package, xres:XMLTransport |



### Attribute

#### @version

*optional*

|          |  |
|----------|--|
| Inhalt   | Enthält die Versionsbezeichnung des eXTra Schemas. <i>Dieses Attribut ist vorgesehen für die Verwendung in Szenarien, in denen Teilstrukturen aus eXTra-Nachrichten herausgelöst und separat verarbeitet werden.</i> |
| Datentyp | xsd:anyURI   |

### Profilierung

| Komponente       | REQ | OPT | FOR | MUL | IGN | VAL |
|------------------|-----|-----|-----|-----|-----|-----|
| xlog:LogSequence | +   | +   | +   |     | +   |     |
| xlog:LogStream   | +   | +   | +   |     | +   |     |

### Beschreibung

xlog:Logging ist das Wurzelement 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.



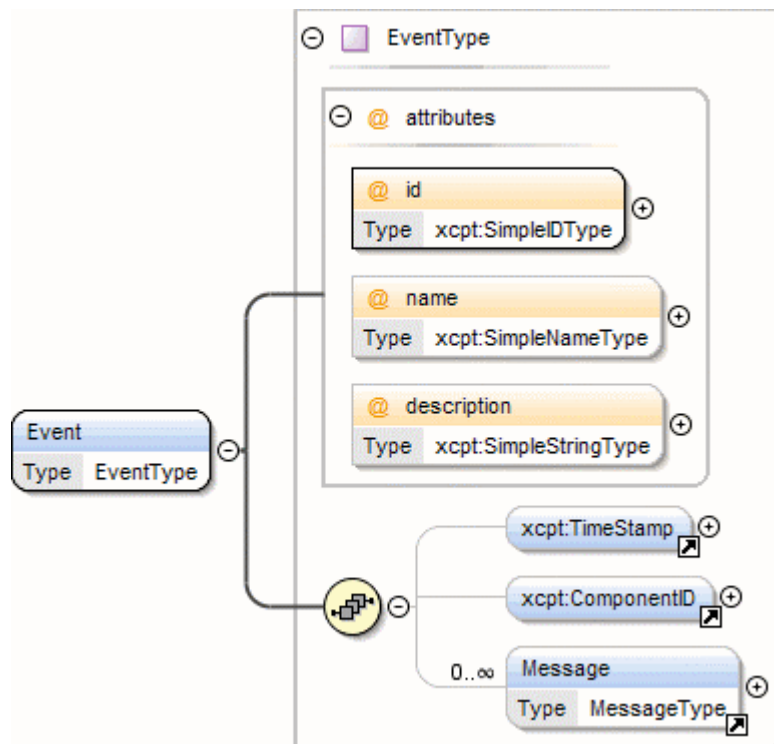
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element Event

### xlog:Event

|             |                                  |
|-------------|----------------------------------|
| Inhalt      | Beschreibt ein Ereignis.         |
| Datentyp    | ##elements                       |
| Benutzt von | xlog:LogSequence, xlog:Operation |



## Beschreibung

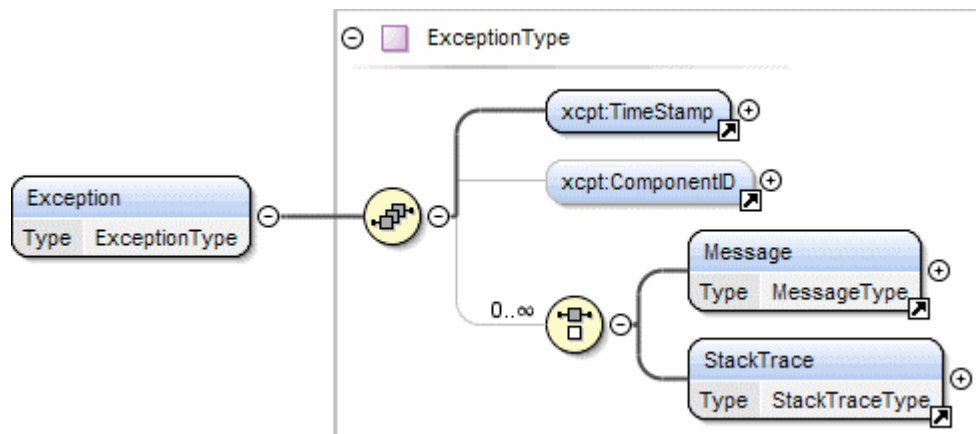
`xlog:Event` beschreibt ein während der Verarbeitung einer Nachricht eingetretenes Ereignis. Das Ereignis ist über eine ID zu referenzieren. Die Klassifikation von Ereignissen und die Vergabe von IDs ist anwendungsspezifisch. Ein Ereignis kann mit einem Zeitstempel und einem Bezug zu einer Komponente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

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element Exception

|             | <b>xlog:Exception</b>            |
|-------------|----------------------------------|
| 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.

### Beispiel

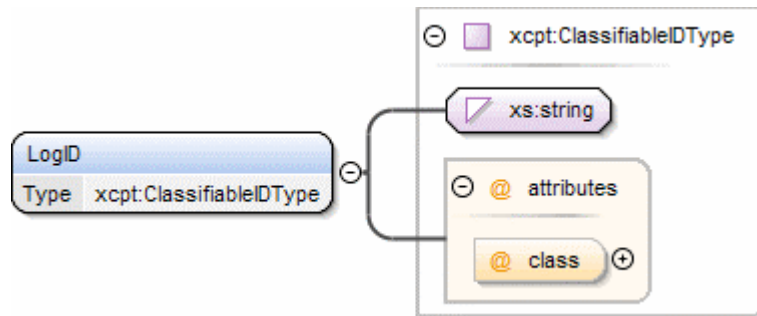
```
<?xml version="1.0" encoding="UTF-8"?>
```



## 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

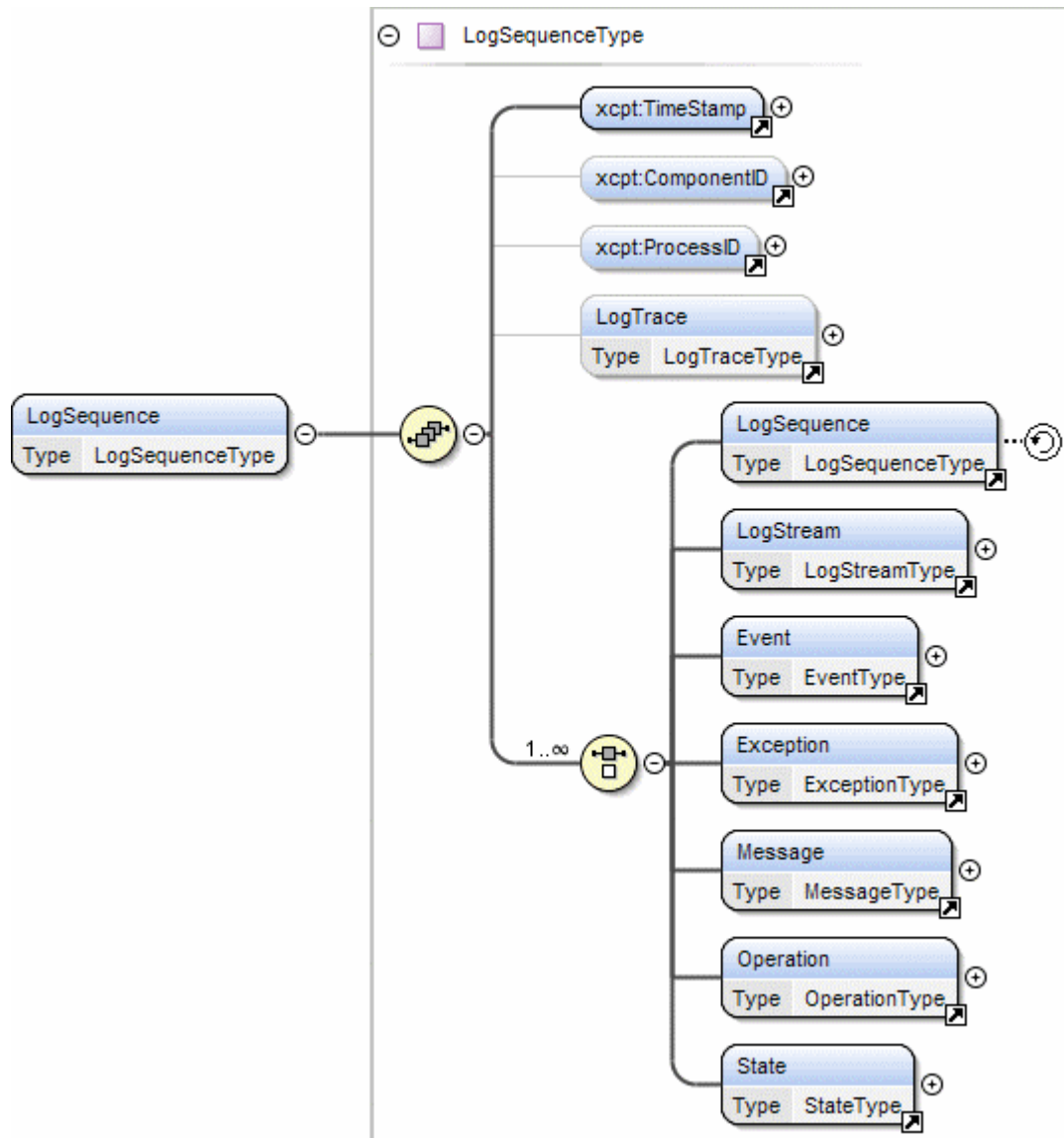
Mit xlog:LogID kann einem xlog:LogSequence bzw. xlog:LogStream Element eine ID zugeordnet werden.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element LogSequence

|             |   |
|-------------|---|
|             | <b>xlog:LogSequence</b>   |
| 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.

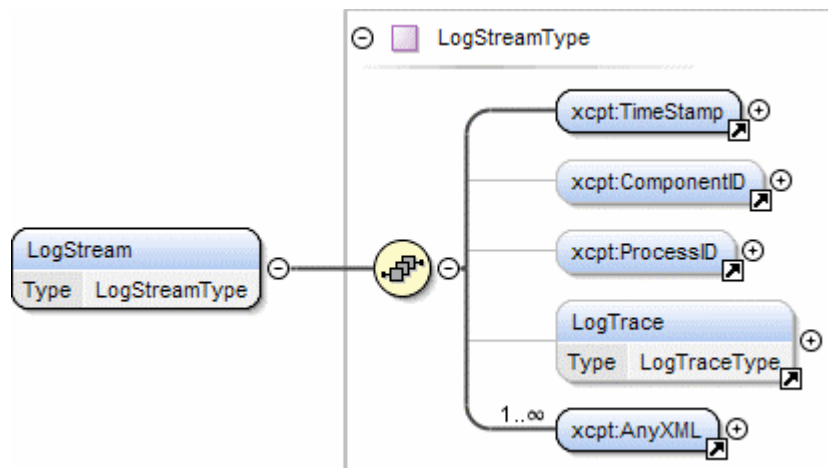


## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element LogStream

|             | <b>xlog:LogStream</b>                    |
|-------------|--|
| 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. Diese Logs sind damit anwendungsspezifisch. Im Ggfs. zu `xlog:LogSequence` ist es mit `xlog:LogStream` ist es deshalb nicht möglich, Logs mit einheitlicher Software qualifiziert auszuwerten.

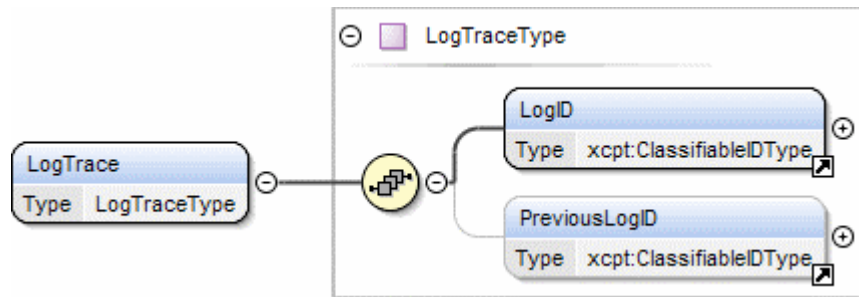
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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.

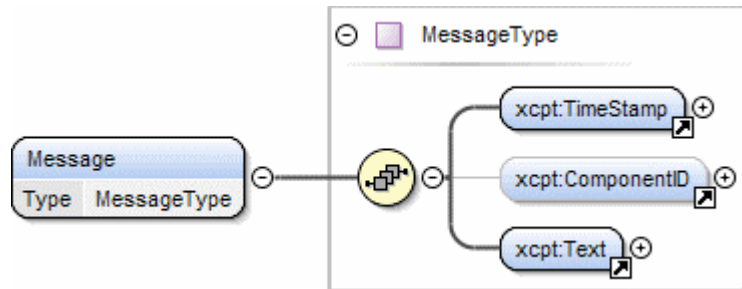
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element Message

### xlog:Message

|             |   |
|-------------|---|
| Inhalt      | Beschreibt eine einem Log-Objekt zugeordnete Nachricht.         |
| Datentyp    | ##elements  |
| Benutzt von | xlog:Event, xlog:Exception, xlog:LogSequence,<br>xlog:Operation |



## Beschreibung

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

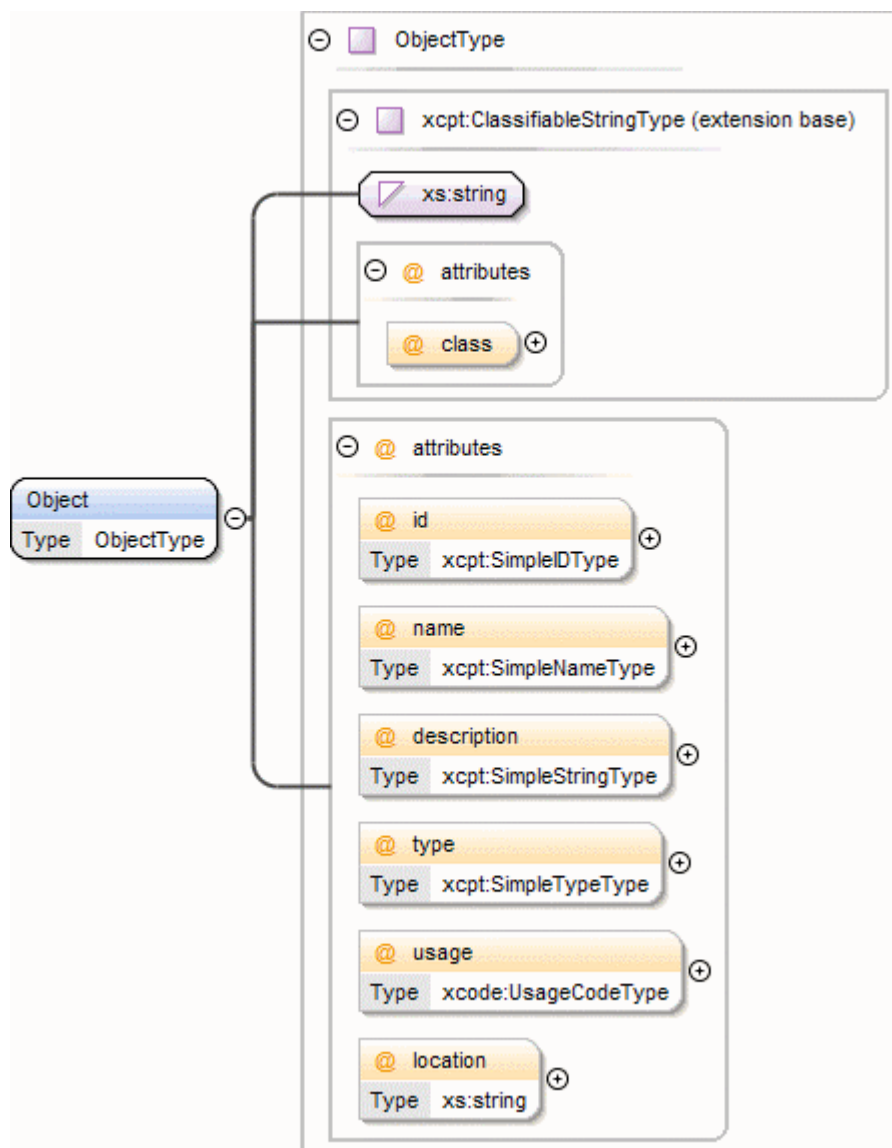
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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                                    |

|                                  | <b>@name</b>   | <i>optional</i> |
|----------------------------------|--|-----------------|
| <i>Inhalt</i><br><i>Datentyp</i> | Der Name des Objekts.<br>xsd:string  |                 |
|                                  | <b>@description</b>  | <i>optional</i> |
| <i>Inhalt</i><br><i>Datentyp</i> | Eine (kurze) Beschreibung Objekts.<br>xsd:string                                       |                 |
|                                  | <b>@type</b>   | <i>optional</i> |
| <i>Inhalt</i><br><i>Datentyp</i> | Eine Angabe zur Typisierung des Objekts.<br>xsd:string                                 |                 |
|                                  | <b>@usage</b>  | <i>optional</i> |
| <i>Inhalt</i><br><i>Datentyp</i> | Eine Angabe zur Art der Verwendung des Objekts.<br>Codeliste <a href="#">UsageCode</a> |                 |
|                                  | <b>@location</b>   | <i>optional</i> |
| <i>Inhalt</i><br><i>Datentyp</i> | Eine Angabe zur Lokalisierung des Objekts.<br>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..

## Beispiel

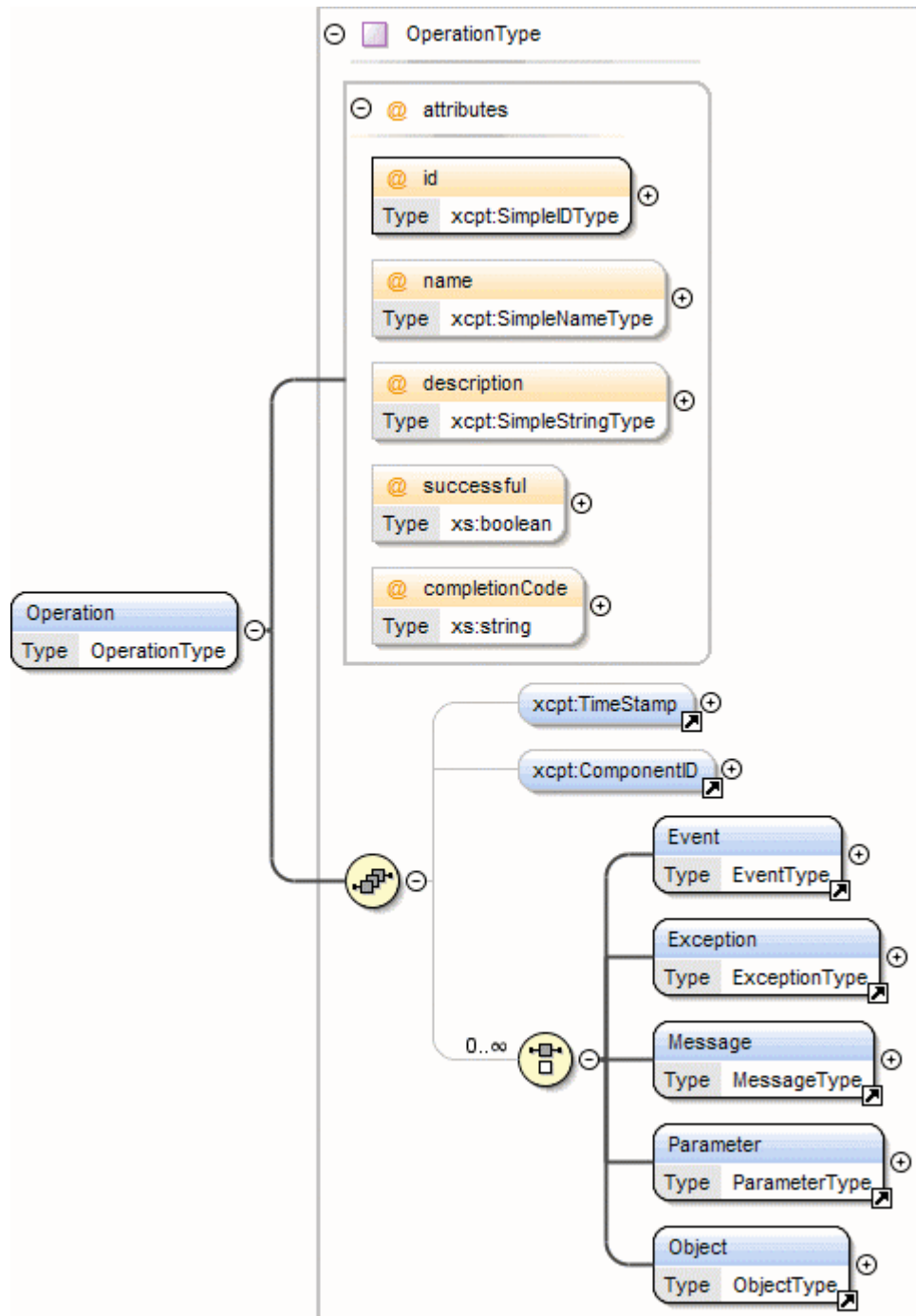
```
<?xml version="1.0" encoding="UTF-8"?>
```



## Element Operation

### xlog:Operation

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



**Attribute**

|                                  | <b>@id</b>   | <i>required</i> |
|----------------------------------|--|-----------------|
| <i>Inhalt</i><br><i>Datentyp</i> | Eine Angabe zur Identifizierung des Objekts.<br>xsd:token  |                 |
|                                  | <b>@name</b>   | <i>optional</i> |
| <i>Inhalt</i><br><i>Datentyp</i> | Der Name des Objekts.<br>xsd:string                        |                 |
|                                  | <b>@description</b>  | <i>optional</i> |
| <i>Inhalt</i><br><i>Datentyp</i> | Eine (kurze) Beschreibung Objekts.<br>xsd:string           |                 |
|                                  | <b>@successful</b>   | <i>optional</i> |
| <i>Inhalt</i><br><i>Datentyp</i> | Zeigt an, ob die Operation erfolgreich war.<br>xsd:boolean |                 |
|                                  | <b>@completionCode</b>                                     | <i>optional</i> |
| <i>Inhalt</i><br><i>Datentyp</i> | Enthält einen Ausführungscode.<br>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 Ausführungscode, 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.

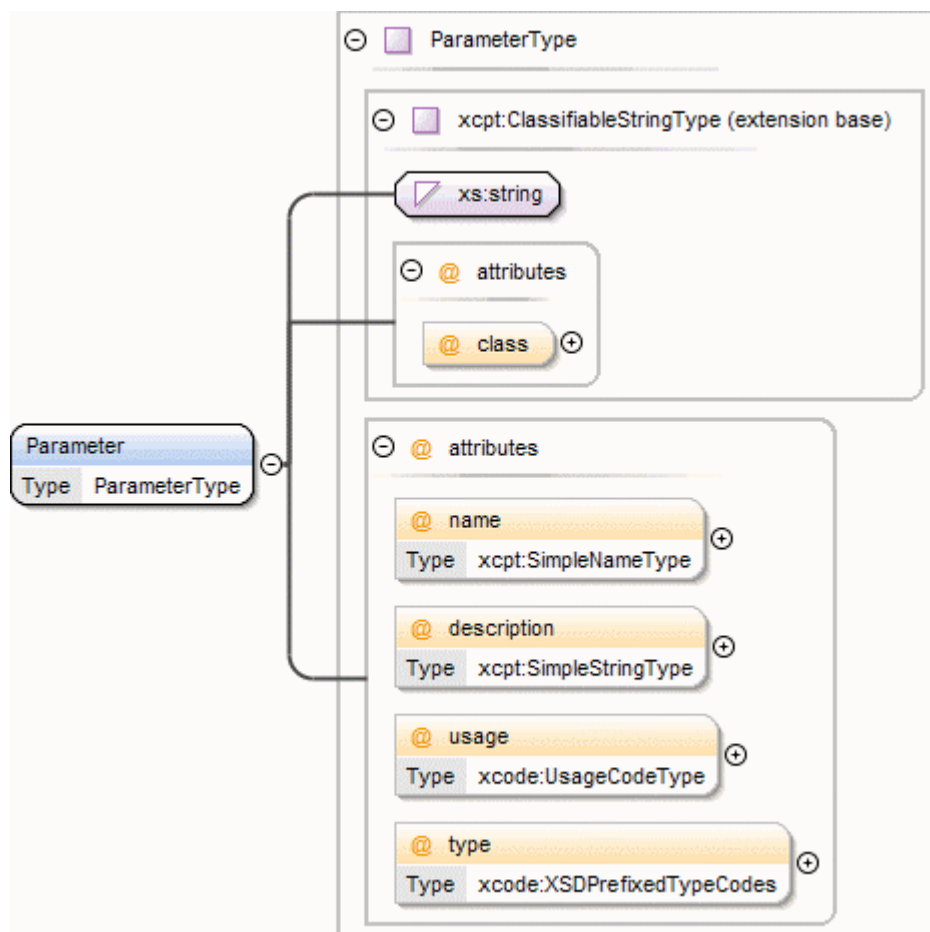
**Beispiel**

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element Parameter

### xlog:Parameter

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



### Attribute

|          |   |                 |
|----------|---|-----------------|
|          | <b>@class</b>                                   | <i>optional</i> |
| Inhalt   | Eine Angabe zur Klassifizierung des Parameters. |                 |
| Datentyp | xsd:string                                      |                 |
|          | <b>@name</b>                                    | <i>optional</i> |
| Inhalt   | Der Name des Parameters.                        |                 |
| Datentyp | xsd:string                                      |                 |
|          | <b>@description</b>                             | <i>optional</i> |
| Inhalt   | Eine (kurze) Beschreibung Parameters.           |                 |
| Datentyp | xsd:string                                      |                 |
|          | <b>@type</b>                                    | <i>optional</i> |

*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.

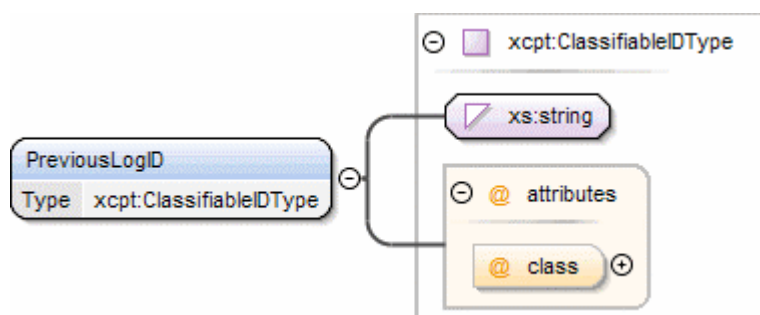
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element *PreviousLogID*

### **xlog:PreviousLogID**

|                    |   |
|--------------------|---|
| <i>Inhalt</i>      | Enthält die ID des historisch vorausgehenden <code>xlog:LogSequence</code> bzw. <code>xlog:LogStream</code> |
| <i>Datentyp</i>    | <code>xsd:string</code>   |
| <i>Benutzt von</i> | <code>xlog:LogTrace</code>  |



## Attribute

### **@class**

*optional*

|                 |   |
|-----------------|---|
| <i>Inhalt</i>   | Eine Angabe zur Klassifizierung der ID. |
| <i>Datentyp</i> | <code>xsd:string</code>                 |

## Beschreibung

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

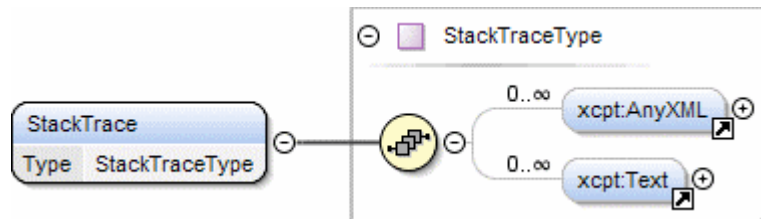
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element StackTrace

### xlog:StackTrace

|             |  |
|-------------|--|
| Inhalt      | Enthält die Repräsentation eines Laufzeitstacks. |
| Datentyp    | ##elements                                       |
| Benutzt von | xlog:Exception                                   |



### Beschreibung

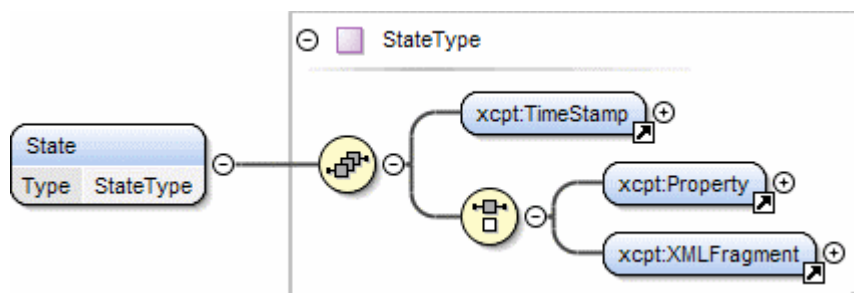
xlog:StackTrace enthält die Repräsentation eines Laufzeitstacks in beliebiger XML-konformer Form.

### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element State

|                    | <b>xlog:State</b>   |
|--------------------|---|
| <i>Inhalt</i>      | Beschreibt einen Zustand als Folge von Name/Wert-Paaren und/oder mit Hilfe von XML-Fragmenten |
| <i>Datentyp</i>    | ##elements  |
| <i>Benutzt von</i> | 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:NamedValue name="connectionAlive" type="xsd:boolean">true</xlog:NamedValue>
  <xlog:NamedValue name="numberOutstandingRequest" type="xsd:integer">12</xlog:NamedValue>
</xlog:State>
```

## **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

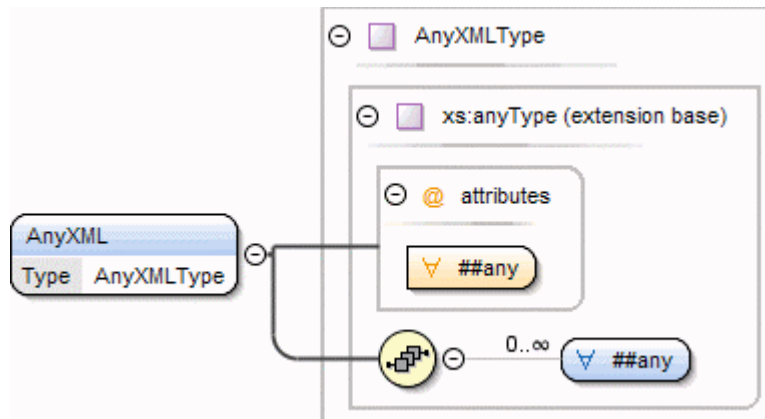
<http://www.extra-standard.de/namespace/components/1>

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



## Element AnyXML

|                    | <b>xcpt: AnyXML</b>                            |
|--------------------|--|
| <i>Inhalt</i>      | Beliebiger XML-konformer Inhalt                |
| <i>Datentyp</i>    | xsd:anyType                                    |
| <i>Benutzt von</i> | 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.

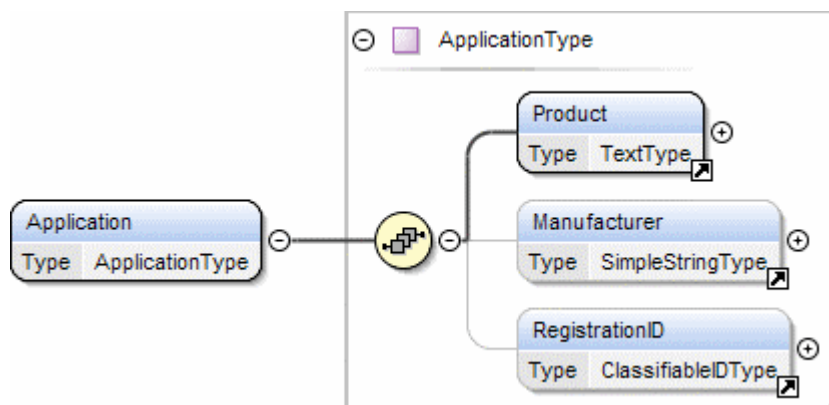
### Beispiel

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

## 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  |

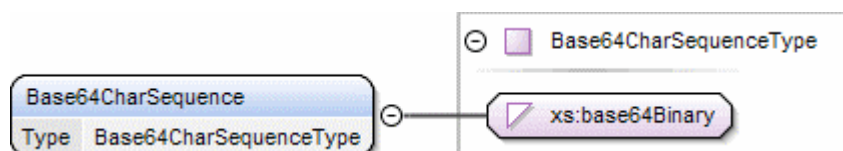
Die Registrierungs-Id dient zur Übertragung von Angaben mit denen eine Registrierung oder Zertifizierung der Anwendung dokumentiert wird.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element Base64CharSequence

|             |                                |
|-------------|--------------------------------|
|             | <b>xcpt:Base64CharSequence</b> |
| Inhalt      | Base64-Zeichenfolge            |
| Datentyp    | xsd:base64Binary               |
| Benutzt von | xcpt:DataType                  |



### Attribute

|          |  |                 |
|----------|--|-----------------|
|          | <b>@numberOfChars</b>                                  | <i>optional</i> |
| Inhalt   | Gibt die Anzahl der Zeichen der Base64-Zeichenfolge an |                 |
| Datentyp | xsd:nonNegativeInteger                                 |                 |

### Beschreibung

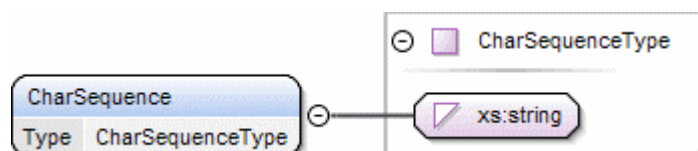
`xcpt:Base64CharSequence` beschränkt den Inhalt des Datencontainers auf eine Base64-konforme Zeichenfolge.

### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element CharSequence

|             |                           |
|-------------|---------------------------|
|             | <b>xcpt:CharSequence</b>  |
| Inhalt      | XML-konforme Zeichenfolge |
| Datentyp    | xsd:string                |
| Benutzt von | xcpt:DataType             |



## Attribute

|          | <b>@numberOfChars</b>                           | <i>optional</i> |
|----------|---|-----------------|
| Inhalt   | Gibt die Anzahl der Zeichen der Zeichenfolge an |                 |
| Datentyp | xsd:nonNegativeInteger                          |                 |

## Beschreibung

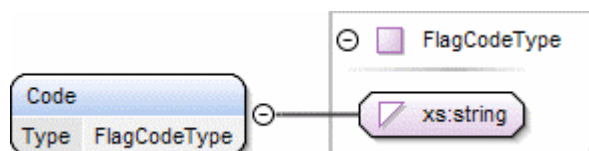
`xcpt:Base64CharSequence` beschränkt den Inhalt des Datencontainers auf eine beliebige XML-konforme Zeichenfolge.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element Code

|                    | <b>xcpt:Code</b>                |
|--------------------|---------------------------------|
| <i>Inhalt</i>      | Ein anwendungsspezifischer Code |
| <i>Datentyp</i>    | xsd:string                      |
| <i>Benutzt von</i> | xcpt:Flag                       |



## Beschreibung

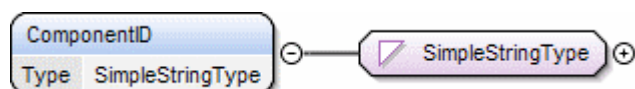
xcpt:Code wird verwendet, um xcpt:Flag einen anwendungsspezifischen Code hinzuzufügen.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element ComponentID

|                    | <b>xcpt:ComponentID</b>   |
|--------------------|---|
| <i>Inhalt</i>      | Anwendungsspezifische Referenz auf eine beliebige Komponente eines eXTra Verarbeitungssystems |
| <i>Datentyp</i>    | xsd:string  |
| <i>Benutzt von</i> | xlog:Event, xlog:Exception, xlog:LogSequence,<br>xlog:LogStream, xlog:Message, xlog:Operation |



## Beschreibung

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

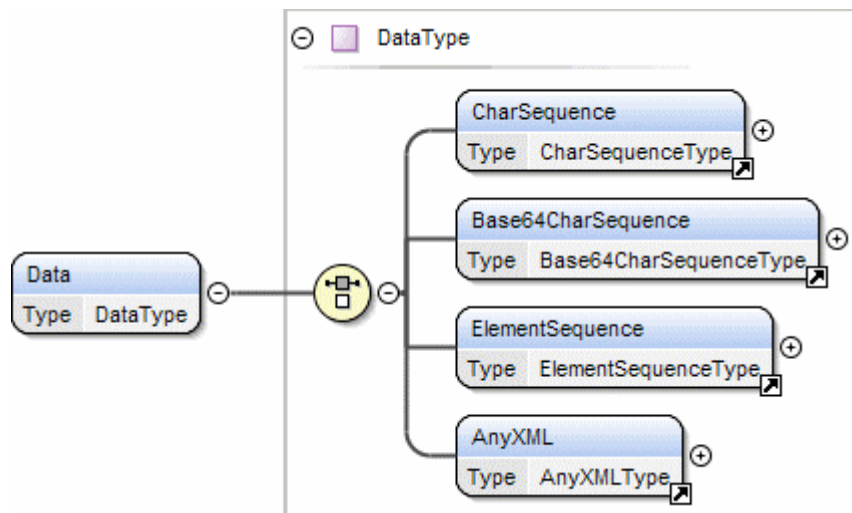
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element Data

### xcpt:Data

|             |   |
|-------------|---|
| Inhalt      | Eine <i>Typelement</i> mit anwendungsspezifischen Daten   |
| Datentyp    | ##elements  |
| Benutzt von | xreq:MessageBody, xreq:PackageBody, xreq:TransportBody,<br>xres:MessageBody, xres:PackageBody, xres:TransportBody |



## Profilierung

| Komponente              | REQ | OPT | FOR | MUL | IGN | VAL |
|-------------------------|-----|-----|-----|-----|-----|-----|
| xcpt:CharSequence       | +   | +   | +   |     | +   |     |
| xcpt:Base64CharSequence | +   | +   | +   |     | +   |     |
| xcpt:ElementSequence    | +   | +   | +   |     | +   |     |
| xcpt:AnyXML             | +   | +   | +   |     | +   |     |

## Beschreibung

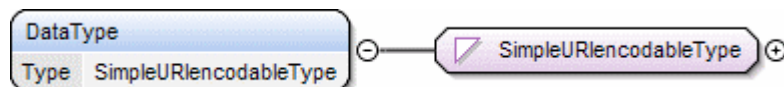
xcpt:Data ist das Wurzelement eines eXtra Datencontainers. Es enthält genau eines von vier Typelementen, die den Datentyp der Nutznachricht formal eingrenzen und die anwendungsspezifischen Nutzdaten enthalten.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element *DataType*

|                    | <b>xcpt:DataType</b>   |
|--------------------|--|
| <i>Inhalt</i>      | Art, Typ oder sonstige kategorisierende oder beschreibende anwendungs-spezifische Angabe über die übermittelten Daten. |
| <i>Datentyp</i>    | xsd:anyURI   |
| <i>Benutzt von</i> | 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.

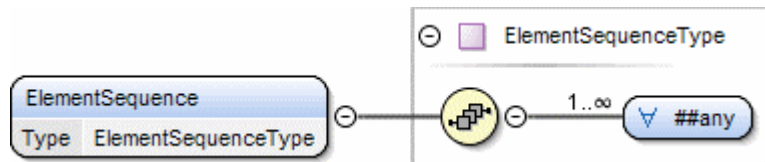
## Beispiel

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



## Element ElementSequence

|             | <b>xcpt:ElementSequence</b>        |
|-------------|------------------------------------|
| 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

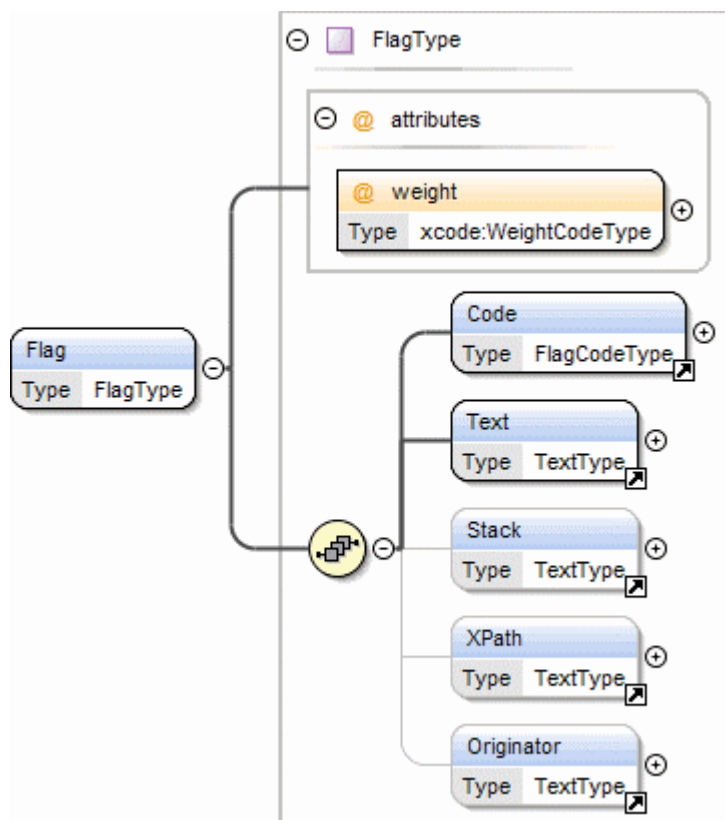
```

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

## 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 <a href="#">WeightCode</a> |

### Beschreibung

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

### Beispiel

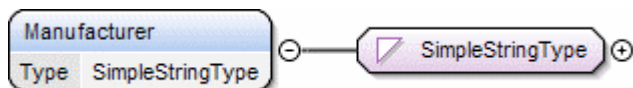
```
<Flag weight="http://www.extra-standard.de/weight/ERROR">
  <Code>X3301-500</Code>
  <Text>Http error 500 was raised</Text>
</Flag>
```

```
<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>
```

## 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 Name oder eine andere Bezeichnung eines Herstellers.

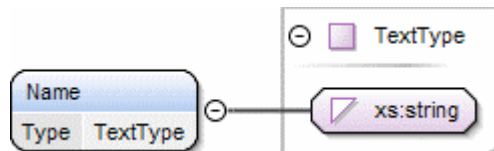
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element Name

### xcpt:Name

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



## Beschreibung

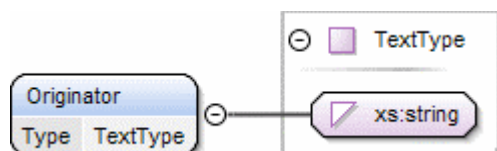
Die Semantik des Namens ergibt sich aus dem Kontext, in dem das Elementes benutzt wird.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element Originator

|                    | <b>xcpt:Originator</b>                 |
|--------------------|--|
| <i>Inhalt</i>      | Angabe über einen Erzeuger/Verursacher |
| <i>Datentyp</i>    | xsd:string                             |
| <i>Benutzt von</i> | 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.

Dieses Element kann auch als Informationsträger bei folgendem Szenario eingesetzt werden:

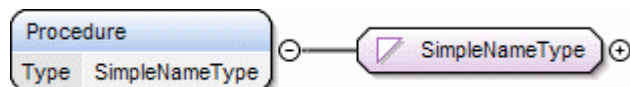
Der Sender hat fachliche Daten mit scenario=request-with-acknowledgement gesendet, deren Annahme der Empfänger mit einem acknowledgement bestätigt hat. Danach erfolgt auf Empfängerseite ein Fehler, z.B. beim Entschlüsseln oder Dekomprimieren, also bevor die Daten an das Fachverfahren weitergegeben werden konnten. Um dem Sender diesen Fehler automatisiert als sog. Acknowledgement2 mitteilen zu können, wird ein Informationsträger für die Identifikation der ursprünglich gesendeten, fehlerhaften Nachricht benötigt. Hierfür bietet sich das Element Originator an; als Identifikation die ResponseID des Acknowledgements der ursprünglichen fehlerhaften Sendung (des Acknowledgement1).

### Beispiel

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

## Element Procedure

|                    | <b>xcpt:Procedure</b>   |
|--------------------|---|
| <i>Inhalt</i>      | Bezeichnung eines (softwaregestützten) technischen oder fachlichen Verfahrens |
| <i>Datentyp</i>    | xsd:string  |
| <i>Benutzt von</i> | xcpt:RequestDetails   |



## Beschreibung

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

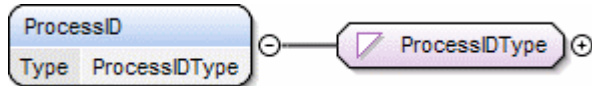
## Beispiel

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

## Element *ProcessID*

### xcpt:ProcessID

|                    |                                     |
|--------------------|-------------------------------------|
| <i>Inhalt</i>      | Id eines technischen (IT-)Prozesses |
| <i>Datentyp</i>    | xsd:string                          |
| <i>Benutzt von</i> | 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

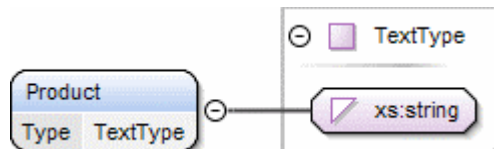
```
<ProcessID>TSN:9A4F</ProcessID>
```



## 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

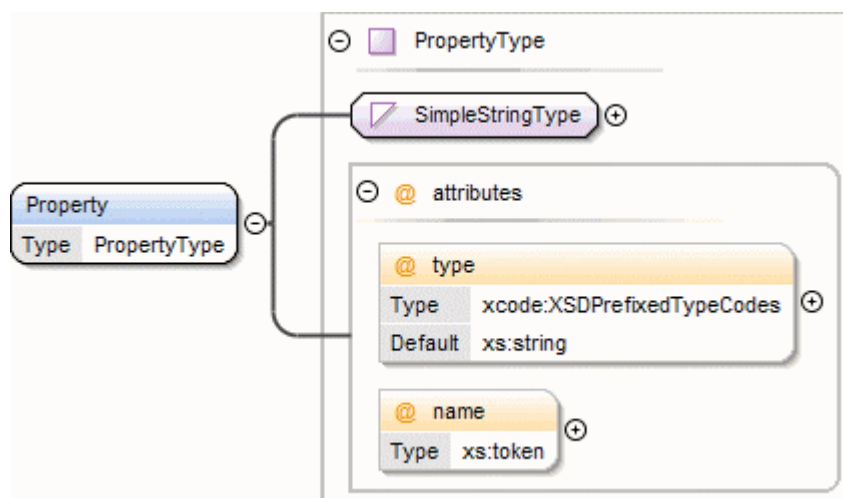
```
<Product>http://www.destatis.de/procduts/KonVert/1.6#Inspector</Product>
```

```
<Product>MySQL 3.1</Product>
```

## 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.

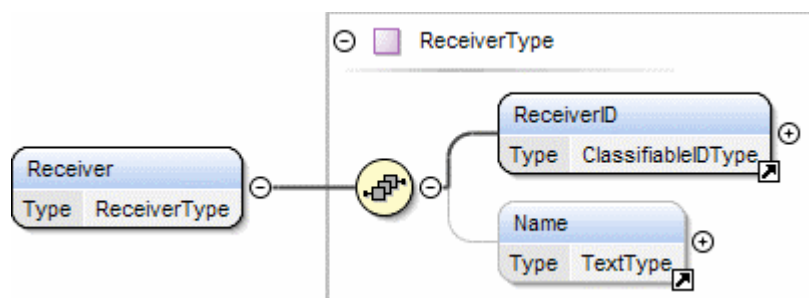
## Beispiel

???

## Element Receiver

### `xcpt:Receiver`

|                    |  |
|--------------------|--|
| <i>Inhalt</i>      | Beschreibung eines Empfängers  |
| <i>Datentyp</i>    | ##elements   |
| <i>Benutzt von</i> | <code>xreq:MessageHeader</code> , <code>xreq:PackageHeader</code> ,<br><code>xreq:TransportHeader</code> ,<br><code>xres:MessageHeader</code> , <code>xres:PackageHeader</code> ,<br><code>xres:TransportHeader</code> |



## 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



**eXtra Transport V1.2**  
**Schnittstellenbeschreibung V1.2.1**

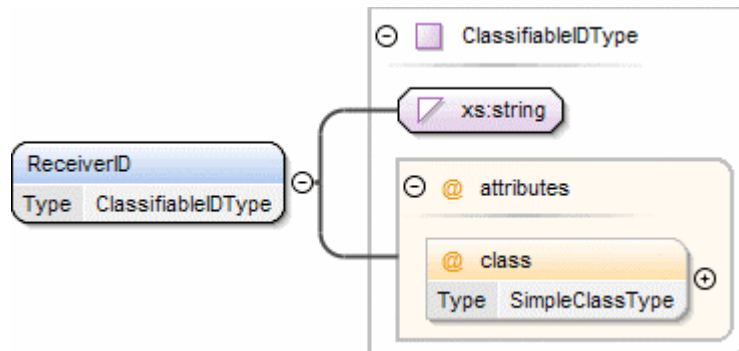
Seite: 131  
Status: **FINAL**  
Stand: **08.09.2010**

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

## 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.

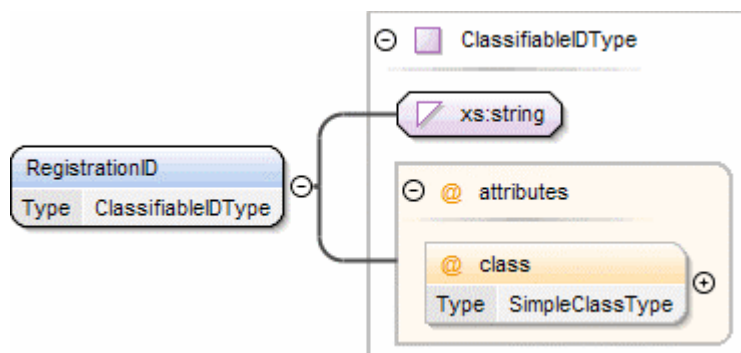
## Beispiel

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

## Element *RegistrationID*

### **xcpt:RegistrationID**

|                    |  |
|--------------------|--|
| <i>Inhalt</i>      | Kennung, mit der ein Objekt registriert oder zertifiziert wurde. |
| <i>Datentyp</i>    | xsd:string   |
| <i>Benutzt von</i> | xcpt:Application   |



## Attribute

### **@class**

*optional*

|                 |                              |
|-----------------|------------------------------|
| <i>Inhalt</i>   | Eine klassifizierende Angabe |
| <i>Datentyp</i> | xsd:anyURI                   |

## Beschreibung

Siehe xcpt:application.

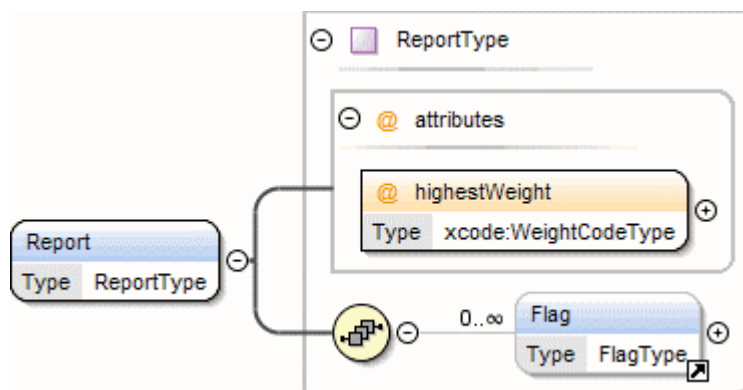
## Beispiel

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

## 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   |
| Wertebereich | Codeliste <a href="#">WeightCode</a>                         |

### 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>
  </Flag>
</Report>
  
```



**eXtra Transport V1.2**  
**Schnittstellenbeschreibung V1.2.1**

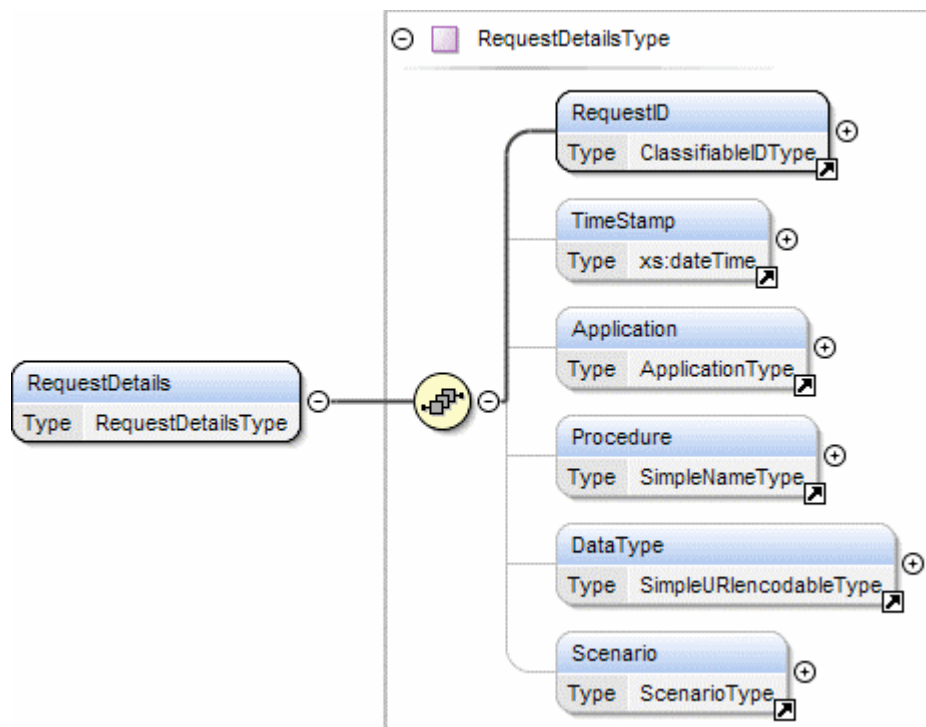
Seite: 135  
Status: **FINAL**  
Stand: **08.09.2010**

```
<Text>Server unavailable 2009-09-30T20:00P3H</Text>  
</Flag>  
</Report>
```

## Element RequestDetails

### xcpt:RequestDetails

|             |   |
|-------------|---|
| Inhalt      | Diverse Request-spezifische Informationen   |
| Datentyp    | ##elements  |
| Benutzt von | xreq:MessageHeader, xreq:PackageHeader,<br>xreq:TransportHeader,<br>xres:MessageHeader, xres:PackageHeader,<br>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          |





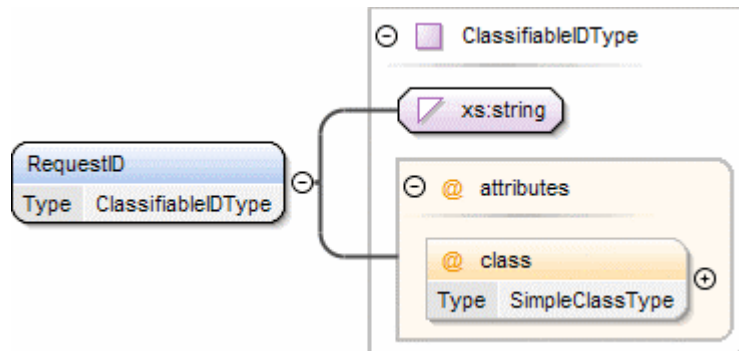
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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.

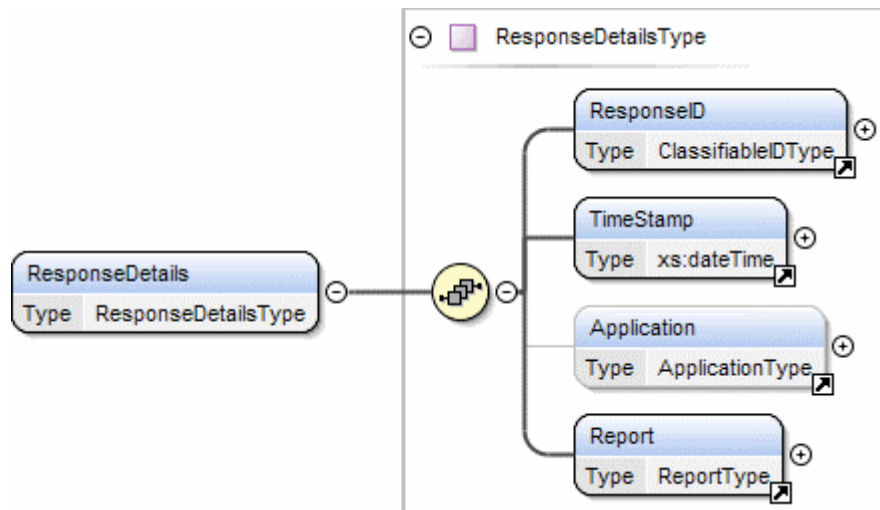
## Beispiel

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

## Element ResponseDetails

### xcpt:ResponseDetails

|             |   |
|-------------|---|
| Inhalt      | Diverse Response-spezifische Informationen                      |
| Datentyp    | ##elements  |
| Benutzt von | xres:MessageHeader, xres:PackageHeader,<br>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 |

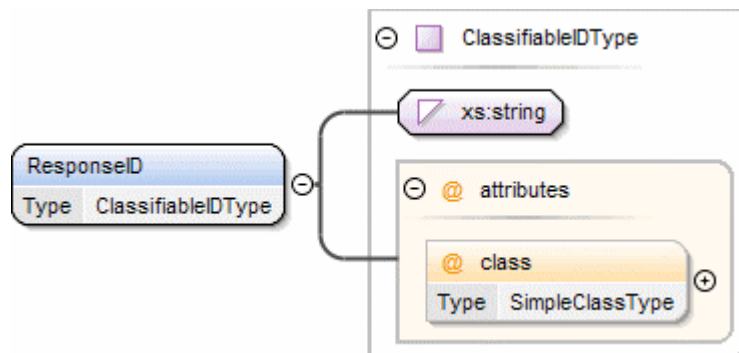
## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 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`.

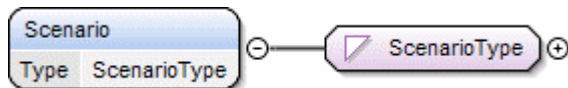
### Beispiel

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

## Element Scenario

### xcpt:Scenario

|             |   |
|-------------|---|
| Inhalt      | Spezifikation des Kommunikationsszenarios |
| Datentyp    | Codeliste <a href="#">ScenarioCode</a>    |
| Benutzt von | xcpt:RequestDetails                       |



## Beschreibung

Gibt das von Absender gewünschte Kommunikationsszenario 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>

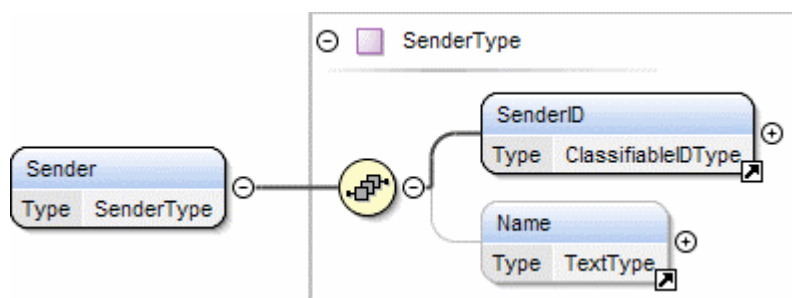
## Beispiel

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

## Element Sender

### xcpt : Sender

|             |   |
|-------------|---|
| Inhalt      | Beschreibung eines Absenders  |
| Datentyp    | ##elements  |
| Benutzt von | xreq:MessageHeader, xreq:PackageHeader,<br>xreq:TransportHeader,<br>xres:MessageHeader, xres:PackageHeader,<br>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.

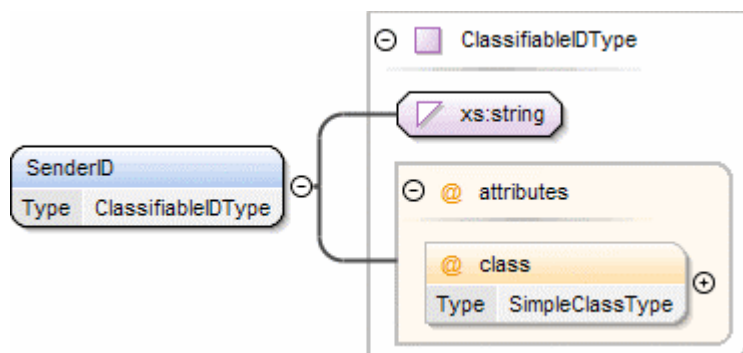
## Beispiel

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

## 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.

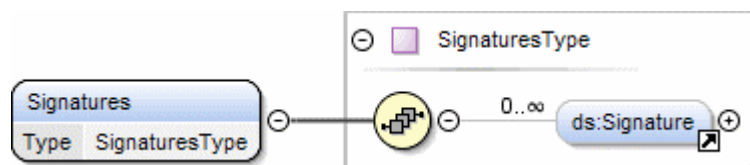
## Beispiel

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

## Element Signatures

### xcpt:Signatures

|             |   |
|-------------|---|
| Inhalt      | Eine beliebige Anzahl von <i>XML Signature</i> konformen Signaturen                             |
| Datentyp    | ##elements  |
| Benutzt von | xreq:Message, xreq:Package, xreq:XMLTransport,<br>xres:Message, xres:Package, xres:XMLTransport |



### Beschreibung

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

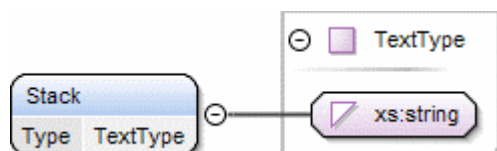
### Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```



## Element Stack

|                    | <b>xcpt:Stack</b>  |
|--------------------|--|
| <i>Inhalt</i>      | Ein Stackprint oder anderes Protokoll einer Aktion einer Anwendung |
| <i>Datentyp</i>    | xsd:string   |
| <i>Benutzt von</i> | xcpt:Flag  |



## Beschreibung

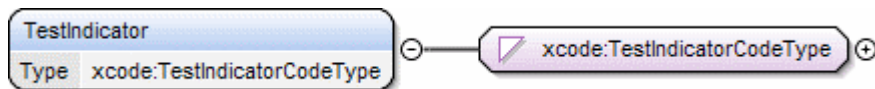
Dieses Element wird benutzt, um Berichtselementen (Flags) Informationen über (insbesondere fehlerhafte) programmatische Abläufe oder Aktionen hinzuzufügen.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Element TestIndicator

|             | <b>xcpt:Testindicator</b>   |
|-------------|---|
| Inhalt      | Indikator für Testfälle und Testfallverarbeitung  |
| Datentyp    | Codeliste <a href="#">TestIndicatorCode</a>   |
| Benutzt von | xreq:MessageHeader, xreq:PackageHeader,<br>xreq:TransportHeader,<br>xres:MessageHeader, xres:PackageHeader,<br>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 Testmarker 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

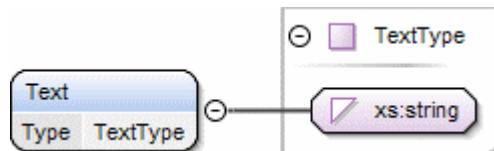
als Folge der Verarbeitung weitere asynchron abzuholende Informationen, wie z.B. Protokolle oder Bescheide, erstellt werden, die ja ebenfalls getestet werden sollten.

### **Beispiel**

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

## Element Text

|                    | <b>xcpt:Text</b>                          |
|--------------------|---|
| <i>Inhalt</i>      | Ein beliebiger Text                       |
| <i>Datentyp</i>    | xsd:string                                |
| <i>Benutzt von</i> | 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>
```

## Element TimeStamp

|                    | <b>xcpt:TimeStamp</b>   |
|--------------------|---|
| <i>Inhalt</i>      | Ein Zeitstempel   |
| <i>Datentyp</i>    | xsd:dateTime  |
| <i>Benutzt von</i> | xcpt:RequestDetails, xcpt:ResponseDetails,<br>xlog:Event, xlog:Exception, xlog:LogSequence,<br>xlog:LogStream, xlog:Message, xlog:Operation, xlog:State |



## Beschreibung

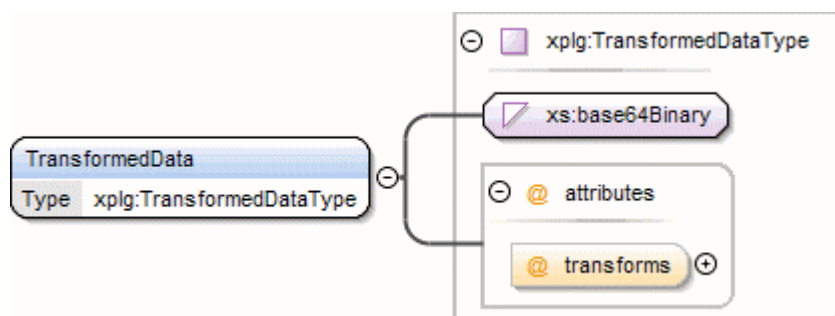
Zeitstempel sind für die Dokumentation einer Datenübermittlung von zentraler Bedeutung und daher an vielen Stellen des eXTra Protokolls Pflicht.

## Beispiel

```
<TimeStamp>2008-09-26T06:09:26</TimeStamp>
```

## Element *TransformedData*

|                    | <b>xcpt:TransformedData</b>   |
|--------------------|---|
| <i>Inhalt</i>      | Daten, die einer Transformation wie z.B. Verschlüsselung unterzogen wurden.                                       |
| <i>Datentyp</i>    | xsd:base64Binary  |
| <i>Benutzt von</i> | xreq:MessageBody, xreq:PackageBody, xreq:TransportBody,<br>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.

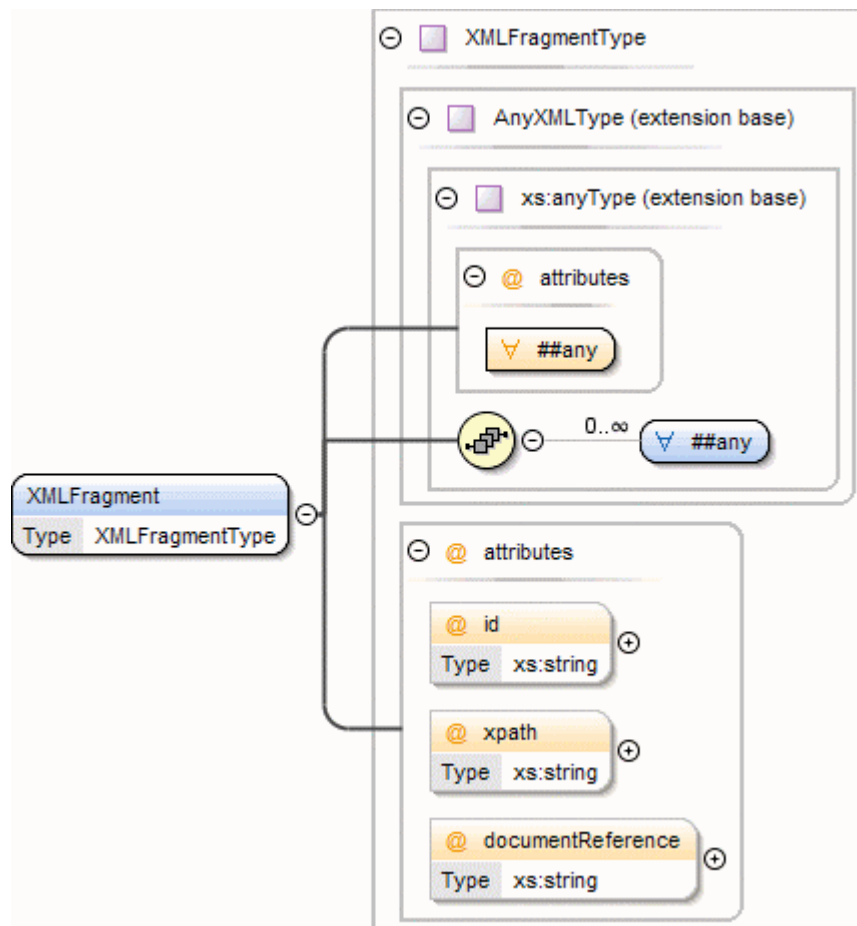
### Beispiel

```
<TimeStamp>2008-09-26T06:09:26</TimeStamp>
```

## Element XMLFragment

**xcpt:XMLFrgament**

|             |   |
|-------------|---|
| Inhalt      | Ein XML Fragment in einem beliebigen Namensraum |
| Datentyp    | ##elements                                      |
| Benutzt von | xlog:State                                      |



### Attribute

|          |                                     |                 |
|----------|-------------------------------------|-----------------|
|          | <b>@id</b>                          | <i>optional</i> |
| Inhalt   | Eine beliebige ID                   |                 |
| Datentyp | xsd:string                          |                 |
|          | <b>@xpath</b>                       | <i>optional</i> |
| Inhalt   | Ein XPath Ausdruck                  |                 |
| Datentyp | xsd:string                          |                 |
|          | <b>@documentReference</b>           | <i>optional</i> |
| Inhalt   | Eine Verweis auf ein (XML) Dokument |                 |
| Datentyp | xsd:string                          |                 |

## **Beschreibung**

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

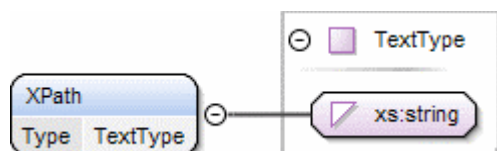
## **Beispiel**

```
<?xml version="1.0" encoding="UTF-8"?>
```



## Element XPath

|                    | <b>xcpt:XPath</b>  |
|--------------------|--------------------|
| <i>Inhalt</i>      | Ein XPath Ausdruck |
| <i>Datentyp</i>    | xsd:string         |
| <i>Benutzt von</i> | 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.

## Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
```

## Ressourcen

### ***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-logging-1.xsd     | Logging-Komponente               |
| eXtra-plugins-1.xsd     | PlugIns                          |
| xenc-schema.xsd         | Komponente XML Encryption        |
| xmldsig-core-schema.xsd | Komponente XML Signature         |

## Literatur

| Kurzname | Quelle   |
|----------|--|
| DSIG     | <i>eXTra Design Guidelines</i> , zu finden unter <a href="http://www.extra-standard.de">www.extra-standard.de</a>  |
| EINF     | <i>Einführung in den eXTra Standard</i> , zu finden unter <a href="http://www.extra-standard.de">www.extra-standard.de</a>   |
| EMSG     | <i>eXTra Standardnachrichten, Schnittstellenbeschreibung</i> , zu finden unter <a href="http://www.extra-standard.de">www.extra-standard.de</a>  |
| IFACE    | <i>eXTra Transport Schnittstellenbeschreibung</i> , zu finden unter <a href="http://www.extra-standard.de">www.extra-standard.de</a>   |
| IMPL     | <i>eXTra Implementierung</i>   |
| KOMP     | <i>eXTra Kompendium</i> , zu finden unter <a href="http://www.extra-standard.de">www.extra-standard.de</a>   |
| RFC2119  | <i>Request for Comments: 2119</i> , S. Bradner, Harvard University, March 1997, <a href="http://www.ietf.org/rfc/rfc2119.txt">http://www.ietf.org/rfc/rfc2119.txt</a>  |
| PROF     | <i>eXTra Profilierung</i> , zu finden unter <a href="http://www.extra-standard.de">www.extra-standard.de</a>   |
| XENC     | <i>XML Encryption</i> , <a href="http://www.w3.org/TR/xmlenc-core/">http://www.w3.org/TR/xmlenc-core/</a>  |
| XML      | <i>XML Recommendation 1.0, 3rd Edition</i> , <a href="http://www.w3.org/XML">http://www.w3.org/XML</a>   |
| XSD      | <i>XML Schema Definition</i> , <a href="http://www.w3.org/TR/xmlschema-0/">http://www.w3.org/TR/xmlschema-0/</a>   |
| XSIG     | <i>XML Signature</i> , <a href="http://www.w3.org/TR/xmldsig-core/">http://www.w3.org/TR/xmldsig-core/</a>   |
| XSL      | <i>XML Stylesheet Language</i> , <a href="http://www.w3.org/TR/1999/REC-xslt-19991116">http://www.w3.org/TR/1999/REC-xslt-19991116</a> , <a href="http://www.w3.org/TR/xslt20/">http://www.w3.org/TR/xslt20/</a> |