

1 Norm 230 – Übertragung von Dateien

2 **Release und Version**

3 Release 1, Version 1, vom 30. Juli 2007

4 **Status**

5 Potentielle Konvention (PN)

6 **Editor**

7 Sören Chittka, VOLKSWOHL BUND (soeren.chittka@volkswohl-bund.de)

8 **Autoren**

9 Sören Chittka, VOLKSWOHL BUND (soeren.chittka@volkswohl-bund.de)

11 **Gegenstand der Norm**

12 Die Norm 230 definiert die technischen Grundlagen für die Übertragung von binären Daten im
13 BiPRO-konformen Nachrichtenaustausch.

14 **Voraussetzung**

15 Norm 200, Release 1 Version 2

16 Norm 270, Release 1 Version 1

17 **Hinweis zur Veröffentlichung**

18 Die anliegend überreichte Norm ist urheberrechtlich für die BiPRO geschützt. Das Dokument
19 wird Ihnen im Rahmen Ihrer Mitgliedschaft bei der BiPRO und damit als Mitglied dieses
20 geschlossenen Empfängerkreises überlassen. Dementsprechend stellt die Überlassung an
21 Sie keine Erstveröffentlichung der Norm dar. Zur Erstveröffentlichung gegenüber Dritten bleibt
22 somit die BiPRO alleine berechtigt. Die Veröffentlichung erfolgt gemäß Norm100. Aufgrund
23 der besonderen Wertigkeit der Normen für die Mitglieder sind wir gehalten, Verstöße
24 gerichtlich zu ahnden.

Inhaltsverzeichnis

26	Norm 230 – Übertragung von Dateien	1
27	Inhaltsverzeichnis.....	2
28	Übertragung von binären Daten.....	3
29	Anforderungen	3
30	Grundlagen	3
31	XOP und MTOM	3
32	Anwendung.....	4
33	Datenmodell für die Dateiübertragung.....	5
34	CT_Datei.....	5
35	Dateityp (ST_Dateityp)	6
36	Dateiname (ST_Dateiname).....	6
37	Komprimierungstyp (ST_Komprimierungstyp)	6
38	Pruefsumme(CT_Pruefsumme)	6
39	Pruefsummetyp (ST_Pruefsummetyp).....	6
40	Pruefsumme (ST_Pruefsumme).....	7
41	Datenlink (ST_URL)	7
42	Daten (xmime:base64binary)	7
43	xmime:base64binary	7
44	contentType (xs:string).....	7
45	Unterstützung von MTOM in aktuellen Frameworks.....	8
46		

47 Übertragung von binären Daten

48 Anforderungen

49 Die Übertragung von binären Daten in XML-basierten Nachrichtenformaten stellt besondere
50 Anforderungen an die Systeme von Service Providern und Consumern. Speziell große
51 Datenmengen, die direkt im XML-Dokument übertragen werden, erzeugen eine erhebliche
52 Last für XML-Parser und können für spürbare Performanceprobleme sorgen.

53 Dieses Dokument stellt eine interoperable Lösung für die Übertragung von großen binären
54 Datenmengen vor und beschreibt darüber hinaus, wie mit kleineren Datenmengen
55 umgegangen werden kann.

56 Grundlagen

57 Das W3C definiert für die Übertragung von Binärdaten in XML-Nachrichten die W3C-
58 Recommendations „XML-binary Optimized Packaging“¹ (kurz XOP) und darauf aufbauend
59 „SOAP Message Transmission Optimization Mechanism“² (kurz MTOM).

60 Im Zusammenhang mit diesen Spezifikationen findet die W3C Working Group Note
61 „Describing Media Content of Binary Data in XML“³ (kurz XML-MIME) Anwendung.

62 Obwohl MTOM grundsätzlich für SOAP-1.2 definiert wurde, lassen sich die beschriebenen
63 Prinzipien identisch auf SOAP-1.1-basierte Nachrichten anwenden.

64 XOP und MTOM

65 Die XOP-Spezifikation beschreibt ein abstraktes Framework zur Überführung eines beliebigen
66 XML-Infosets in eine optimierte serialisierte Form. Bei dieser Serialisierung werden die Inhalte
67 des Infosets nicht inline übertragen, sondern in ein sogenanntes XOP-Package geschrieben.
68 Die binären Daten werden dabei extrahiert und an ihrer Stelle ein Zeiger auf den Teil des
69 XOP-Packages eingebettet, der den extrahierten Inhalt enthält. Die extrahierten binären Daten
70 sind nach diesem Vorgang nicht mehr base64-encodiert, so dass das Datenvolumen sinkt.
71 Die Semantik des XML-Infosets bleibt davon unbeeinträchtigt, d.h. eine verarbeitende Anwendung
72 MUSS die Daten als base64-encodiert und inline ansehen, z.B. beim Erzeugen von
73 Signaturen oder Verschlüsselungen.

74 MTOM nutzt nun dieses Framework und definiert eine konkrete Implementierung für SOAP-
75 Nachrichten und eine Serialisierung in das MIME Multipart/Related-Format. Dabei wird jede
76 binäre Datei innerhalb der gesamten SOAP-Nachricht – nicht nur in der Payload – in ein
77 eigenes „multipart/related“-Paket einer MIME-Nachricht verpackt und an der entsprechenden

¹ XOP: <http://www.w3.org/TR/xop10/>
² MTOM: <http://www.w3.org/TR/soap12-mtom/>
³ XML-MIME: <http://www.w3.org/TR/xml-media-types/>

78 Stelle der SOAP-Nachricht ein Verweis auf die jeweilige Content-ID dieses Pakets eingefügt.

79 Da jedes „multipart/related“-Paket einen eigenen MIME-Content-Type haben kann, wird
 80 empfohlen, das Element im originalen XML-Infoset mit dem Attribute xmime:contentType
 81 auszuzeichnen.

82 Das erzeugte Gesamtpaket ist völlig kompatibel zu „SOAP with Attachments“⁴ (kurz SwA).

83 Anwendung

84 Ein BiPRO-konformer Service KANN MTOM verwenden, um Binärdateien zu versenden.
 85 Wenn das der Fall sein sollte, MUSS **jede** Nachricht, die der Service versendet, in MIME
 86 Multipart/Related-Pakete serialisiert werden, auch dann, wenn sie keinerlei Binärdaten enthält.
 87 Darüber hinaus MUSS ein Service, der MTOM nutzt, dies durch seine Policy bekannt geben.
 88 Das geschieht durch Verwendung der W3C-Submission „WS-MTOMPolic“⁵:

89 Verwendung von WS-MTOMPolic

```
90 <wsp:Policy wsu:Id="{x1}" xmlns:wsp="{x2}">
91   <wsoma:OptimizedMimeSerialization xmlns:wsoma="{x3}" />
92   <!-- weitere Policies gemäß Norm 270 -->
93 </wsp:Policy>
```

Variable	Wert
{X1}	Namespace WS-Utility: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
{X2}	Namespace WS-Policy: http://schemas.xmlsoap.org/ws/2004/09/policy
{X3}	Namespace WS-MTOMPolic: http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization

95

96 Zudem MUSS ein Service, der MTOM unterstützt, Anfragen sowohl im „multipart/related“-
 97 Format als auch in Standard-SOAP-Nachrichten akzeptieren.

98 Jedes BiPRO-konforme Projekt SOLLTE entscheiden und dokumentieren, ob die
 99 projektspezifischen Webservices MTOM nutzen dürfen bzw. müssen.

100 Für Projekte, in denen nicht anzunehmen ist, dass große binäre Dateien transferiert werden
 101 müssen, ist es durchaus sinnvoll MTOM **nicht** einzusetzen, da genau wie beim Parsen von
 102 Nachrichten, die Binärdaten enthalten, auch bei der Wandlung des XML-Infosets in eine

⁴ SwA: <http://www.w3.org/TR/SOAP-attachments>

⁵ WS-MTOMPolic: <http://www.w3.org/Submission/2006/SUBM-WS-MTOMPolic-20061101/>

103 optimierte Serialisierung zusätzlicher Rechenaufwand auftritt. In diesem Fall MÜSSEN die
104 Daten base64-encodiert, inline in der SOAP-Nachricht übertragen werden.

105 Es kann keine generelle Aussage darüber getroffen werden, ab welcher Dateigröße es
106 sinnvoll wird, MTOM zu nutzen. Exakte Ergebnisse können nur durch Benchmarking der
107 jeweiligen Anwendung ermittelt werden. Als grober Richtwert, ab dem MTOM verwendet
108 werden sollte, kann eine Dateigröße von über 2 MB angesehen werden.

109 Jedes Projekt MUSS darüber hinaus entscheiden, ob das verwendete Transportprotokoll
110 (meistens TCP/IP) als zuverlässig genug angesehen werden kann, um beliebig große Dateien
111 zu versenden. Ist das nicht der Fall, so MUSS auf **fachlicher** Ebene dafür Sorge getragen
112 werden, dass Informationen in mehreren kleineren Dateien bereitgestellt werden, die einzeln
113 abgeholt werden können. Jede einzelne Datei SOLLTE durch den Consumer bestätigt oder
114 erneut abgeholt werden können, um zu gewährleisten, dass nur fehlerhafte oder fehlende
115 Teile mehrfach übertragen werden.

116 Datenmodell für die Dateiübertragung

117 Unabhängig davon, ob zur Nachrichtenübermittlung MTOM eingesetzt wird oder nicht, MUSS
118 folgendes Datenmodell zur Repräsentation von Dateien in BiPRO-konformen Nachrichten
119 verwendet werden:

120 CT_Datei

121 Datenmodell für Datei-Objekte (Kommentare ausgelassen)

```
122 <complexType name="CT_Datei" final="restriction" abstract="false">
123   <complexContent>
124     <extension base="bipro:CT_Objekt">
125       <sequence>
126         <element name="Dateityp" type="bipro:ST_Dateityp" />
127         <element name="Dateiname" type="bipro:ST_Dateiname" />
128         <element name="Komprimierungstyp" type="bipro:ST_Komprimierungstyp"
129           minOccurs="0"
130         />
131         <element name="Pruefsumme" type="bipro:CT_Pruefsumme" minOccurs="0" />
132
133         <!-- Eines der folgenden Element MUSS verwendet werden -->
134         <element name="Datenlink" type="bipro:ST_URL" minOccurs="0" />
135         <element name="Daten" type="xmime:base64binary" minOccurs="0" />
136       </sequence>
137     </extension>
138   </complexContent>
139 </complexType>
140 <complexType name="CT_Pruefsumme">
141   <simpleContent>
142     <extension base="string">
143       <attribute name="Typ" type="bipro:ST_Pruefsummetyp" use="required" />

```

144 </extension>
145 </simpleContent>
146 </complexType>

147 Der ComplexType CT_Datei stellt eine allgemeine Klasse zur Beschreibung einer einzelnen
148 Datei zur Verfügung. Dabei stehen die folgende Subelemente zur Verfügung:

149 ***Dateityp (ST_Dateityp)***

150 Der Dateityp stellt eine grobe Charakterisierung der Datei dar. Es sind die Werte „Dokument“
151 und „Nettodaten“ erlaubt. Es MUSS genau ein Wert vorgegeben werden.

152 ***Dateiname (ST_Dateiname)***

153 Das Element Dateiname MUSS belegt sein. Der Dateiname SOLLTE ein sprechender Name
154 sein und darf zwischen 1 und 255 Zeichen lang sein.

155 ***Komprimierungstyp (ST_Komprimierungstyp)***

156 Der Komprimierungstyp gibt an, ob und in welcher Form die binären Daten komprimiert
157 wurden. Das Element ist optional. Wird es nicht angegeben, so sind die Daten nicht
158 komprimiert. Der einzige zulässige Wert bei Verwendung des Elements ist derzeit „zip“ und
159 steht für eine Komprimierung mit dem ZIP-Algorithmus.

160 ***Pruefsumme(CT_Pruefsumme)***

161 Dieses Element ist optional. Wird es nicht angegeben, so wurde keine Prüfsumme über die
162 binären Daten angelegt, ansonsten enthält es die Prüfsumme, die nach dem Algorithmus
163 berechnet wird, der im Attribut „Typ“ angegeben wird.

164 Das Attribut „Typ“ enthält einen der Werte „MD5“ oder „SHA1“, wie sie im Datentyp
165 ST_PruefsummeTyp definiert sind. Diese Attribut MUSS belegt werden, wenn es ein Element
166 „Pruefsumme“ gibt.

167 ***Datenlink (ST_URL)***

168 Der Datenlink stellt einen URL auf die Daten, die durch dieses CT_Datei-Objekt repräsentiert
169 werden, dar. In diesem Fall sind die Binärdaten nicht direkt im Objekt hinterlegt, sondern
170 können von der angegebenen URL abgeholt werden. Dieses Element ist optional, aber
171 mindestens dieses oder das Element „Daten“ MUSS belegt sein.

172 ***Daten (xmime:base64binary)***

173 Dieses Element enthält die eigentlichen binären Daten (vgl. xmime:base64binary). Das
174 Element ist optional, aber mindestens dieses oder das Element „Datenlink“ MUSS belegt sein.

175 **xmime:base64binary**

176 **Auszug aus <http://www.w3.org/2005/05/xmlmime>**

```
177 <xs:attribute name="contentType">
178   <xs:simpleType>
179     <xs:restriction base="xs:string">
180       <xs:minLength value="3" />
181     </xs:restriction>
182   </xs:simpleType>
183 </xs:attribute>
184 <xs:complexType name="base64binary">
185   <xs:simpleContent>
186     <xs:extension base="xs:base64binary">
187       <xs:attribute ref="xmime:contentType" />
188     </xs:extension>
189   </xs:simpleContent>
190 </xs:complexType>
```

191 Der ComplexType xmime:base64binary stellt einen Container für die eigentlichen binären
192 Daten zur Verfügung. Er enthält die base64-encodierte Version der zu übertragenden Daten.
193 Genau diese Daten werden bei Verwendung von MTOM extrahiert und als Attachment
194 versendet.

195 ***contentType (xs:string)***

196 Das Attribut contentType beschreibt das Format der Daten in Form eines eindeutigen MIME-
197 Typs. Der Wert MUSS gesetzt werden.

198

199 **Unterstützung von MTOM in aktuellen Frameworks**

200 Von folgenden Frameworks ist bekannt, dass sie MTOM unterstützen:

- 201 • Windows Communication Framework (WCF) ab Version 3.0
- 202 • Apache CXF
- 203 • Apache AXIS2
- 204 • SUN JAX-WS RI 2.0

205 In allen bekannten Frameworks lässt sich MTOM über einen einfachen
206 Konfigurationsparameter aktivieren.

207 Da mittlerweile auch das WS-I MTOM als interoperablen Standard empfiehlt, ist anzunehmen,
208 dass auch weitere Frameworks, die sich noch in aktiver Entwicklung befinden, demnächst
209 MTOM-fähig werden.