



<b>XML-Baukasten für Behördenkommunikation, Basistypen</b>		<b>Konvention</b>
		<b>xml-baukasten 1.0.0</b>
		<b>Empfehlung</b>
Kurzbeschreibung	Diese Spezifikation definiert das Rahmenwerk für einen Baukasten von Basistypen der Behördenkommunikation im E-Government mittels W3C XML Schema.	
Autor(en):	Herbert Pacnik Peter Reichstädter Peter Lieber Franz-Josef Herpers Michael Liehmann Maria Wimmer	Projektteam / Arbeitsgruppe  AG Kommunikationsarchitektur (Q-KA)  BKA / Bereich IKT-Strategie

Stelle	Vorgelegt am	Angenommen am	Abgelehnt am
IKT Board	18.4.2006	2.5.2006	
Länder	18.4.2006	2.5.2006	
Gemeindebund	18.4.2006	2.5.2006	
Städtebund	18.4.2006	2.5.2006	

## XML-Baukasten

### Inhaltsverzeichnis

(1)	Einleitung .....	3
(2)	Organisatorische Prozesse .....	4
(3)	Grundlegende Design-Prinzipien .....	5
	(3.1) Sprache .....	5
	(3.2) Namenskonventionen .....	5
	(3.3) Wiederverwendbarkeit.....	5
	(3.4) Namensraumbezeichner.....	5
	(3.5) Versionierung .....	6
(4)	Verwendung des Basistypenschemas.....	7
	(4.1) Import des Basistypenschemas .....	7
	(4.2) Referenzierung eines Elementtyps .....	7
	(4.3) Referenzierung eines komplexen Typs.....	7
(5)	Richtlinien für Basis-Komponenten.....	8
	(5.1) Empfohlene Elemente.....	8
	5.1.1. Verfahren.....	8
	5.1.2. Weitere Elemente .....	8
	(5.2) Empfohlene Typen .....	8
	5.2.1. Empfänger .....	9
	5.2.2. Kontakte .....	9
	5.2.3. Beilagen.....	10
(6)	Referenzen.....	11

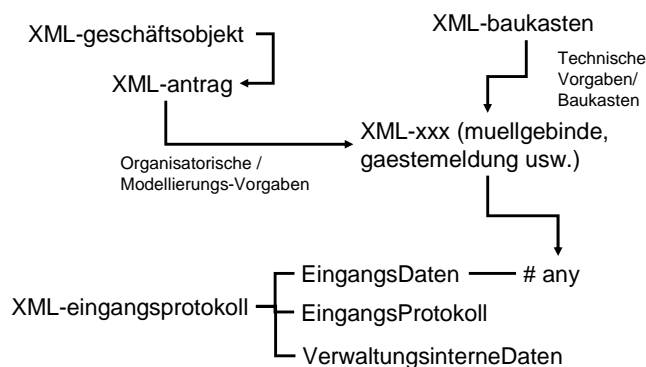
## (1) Einleitung

Das vorliegende Dokument spezifiziert die Verwendung des beiliegenden Schemagerüsts für zu definierende Basistypen der Behördenkommunikation im E-Government. Es wird ein Rahmenwerk zur Definition von Basiselementen und Basistypen bereitgestellt und der Umgang damit, sowie der Erweiterungsprozess desselben beschrieben. Die Definition des Rahmenwerks hält sich an die im *W3C XML Schema (WXS)* (s. **[wxs-1]** und **[wxs-2]**) vorgegebenen Standards.

Ziel dieses Dokuments und des Schemas ist es, einen flexiblen Standardelemente-Baukasten für beliebige E-Government-Schemata zur Verfügung zu stellen. Ein solches Bausteinsystem gewährt die nötige Flexibilität beim Erstellen von spezifischen Schemata, die starre Schemakonstrukte nicht bieten können. Gleichzeitig dient es der Durchsetzung einer einheitlichen Struktur bei den wichtigsten Bausteinen, den Basistypen.

E-Government Applikationsdesigner und Applikationsentwickler können und sollen diese standardisierten Bausteine verwenden und die nachfolgenden Erläuterungen beachten.

Bevor die einzelnen Basistypen erläutert werden, folgt als Nächstes in einem organisatorischen Teil eine Beschreibung des Erweiterungsprozesses der Basiskomponenten. Danach werden grundlegende Design-Prinzipien von Schemata und Richtlinien für die Verwendung der Basistypen formuliert.



## (2) Organisatorische Prozesse

Das Baukastensystem der XML-Basiskomponenten ist vom Prinzip her ein lebendes Modell und folglich veränderlich. Die Prozesse, wie der Baukasten der Basiskomponenten erweitert werden kann, werden hier folgend beschrieben.

Die Hauptverantwortlichkeit für Änderungen liegt bei der Leitung der Arbeitsgruppe Kommarch.

Änderungen am Schema erfolgen nur nach Abstimmung der Arbeitsgruppenleitung mit den Teilnehmern der AG Kommarch. Eine Beschlussfassung per E-Mail ist zulässig.

Eine Änderung kann ein Neueintrag, eine Überarbeitung und das Herausstreichen einer Komponente sein.

Die zusätzlichen Eintragungen in das Schema müssen den hier in weiterer Folge beschriebenen Grundsätzen gehorchen.

Ein Antrag auf Schemaänderung muss an die Leitung der AG Kommarch herangetragen werden.

Der Antrag muss aus drei Einzelteilen bestehen:

- dem ausgeformten Schemateil, welcher den vorliegenden Grundsätzen gehorcht,
- einer erklärenden Beschreibung, die in weiterer Folge das vorliegende Dokument ergänzt,
- einer zusätzlichen textuellen Klärung der Einsatzmöglichkeiten und der Anwendungsfälle, die durch den Einsatz der neuen Basiskomponenten erleichtert werden.

Nach erfolgter Änderung des Schemas muss dieses mit neuer Versionsnummer versehen auf dem Reference Server abgelegt werden.

## (3) Grundlegende Design-Prinzipien

### (3.1) Sprache

Als Sprache für die Bezeichnung der Schemakomponenten kann Deutsch und Englisch gewählt werden. Deutsch ist dann zu bevorzugen, wenn die Begriffe in ihrer deutschen Form spezifisch für die österreichische Behördenkommunikation sind.

### (3.2) Namenskonventionen

Elementtypnamen sollten der *UpperCamelCase-Konvention* (UCC-Konvention) folgen. Attributnamen sollten der *lowerCamelCase-Konvention* (LCC-Konvention) folgen

UCC schreibt den ersten Buchstaben eines jeden Wortes groß und konkateniert dann die Wörter zu einer Zeichenkette: z.B. `GewerbeAnmeldung`. LCC folgt derselben Regel mit der Ausnahme, dass der erste Buchstabe des ersten Wortes klein geschrieben wird, z.B.: `gewerbeId`.

Namen von komplexen Datentypen sollten den Namen des zugehörigen globalen Elements plus dem Suffix „Typ“ tragen und die UCC-Konvention nutzen, z.B. `BeilagenTyp`.

### (3.3) Wiederverwendbarkeit

Das beiliegende Schema ist auf maximale Wiederverwendbarkeit ausgelegt und daher gemäß einer Kombination aus dem so genannten *Venetian Blind Model* und dem *Salami Slice Model* aufgebaut. Dieses Modell darf auch beim Hinzufügen weiterer Bausteine nicht durchbrochen werden. Das bedeutet konkret:

- Alle definierten Typen sind benannt, folglich global deklariert.
- Alle deklarierten Elementtypen sind global deklariert.

Es ist natürlich möglich, die Kindelemente komplexer Basistypen lokal zu deklarieren, wenn diese nicht als eigene Basistypen zur Verfügung stehen sollen.

Auf diese Weise wird der höchstmögliche Grad an Wiederverwendbarkeit zugelassen. Autoren von Spezifikationen können sich der von dieser Spezifikation zur Verfügung gestellten Basistypen dann auf zwei Arten bedienen:

- Referenzierung der global deklarierten Elementtypen in ihren eigenen Inhaltsmodellen.
- Definition eigener Elementtypen und Zuweisung der global definierten Typen.

Eine detaillierte Anleitung zur Einbindung und Verwendung des Basistypenschemas dieser Spezifikation in ein zu definierendes Schema findet sich im Kapitel „Verwendung des Basistypenschemas“.

### (3.4) Namensraumbezeichner

Der Namensraumbezeichner (s. **[nsp-1.0]**) für den Zielnamensraum des Basistypenschemas dieser Spezifikation lautet:

<http://reference.e-government.gv.at/namespace/xml-baukasten/1#>

Er stellt eine URI nach dem http-Schema dar (**[uri]** vs. **[url]**). Als Domainbezeichner ist der Reference-Server (`reference.e-`

government.gv.at) angeführt.

Es folgt das Schlüsselwort `namespace` und das Kürzel des Standards (`xml-baukasten`) als Pfadangaben. Hier wird aufgrund der vorgeschlagenen Versionierungsstrategie (siehe folgendes Kapitel (3.5)) die erste Stelle der Versionsnummer (*Major Version*) angehängt.

Alle Elementtypen, inkl. der lokal deklarierten, sind qualifiziert, Attribute nicht.

### **(3.5) Versionierung**

Die Versionierung des Schemas erfolgt über das `version`-Attribut des `xs:schema`-Elements. Die Versionsnummer korreliert mit der des vorliegenden Dokuments.

Änderungen in der ersten Stelle der Versionsnummer sollten eine neue Version anzeigen, die nicht kompatibel zur alten Version sein muss (*Major Version*). Änderungen der zweiten und dritten Stelle der Versionsnummer zeigen neue Versionen an, die kompatibel zur Vorgängerversion sind (*Minor Version*). Dies entspricht im Wesentlichen dem in **[rm]** vorgeschlagenen Vorgehen. Wenn beispielsweise ein neuer Basistyp hinzugefügt wird, erfolgt eine Änderung in der zweiten Stelle der Versionsnummer.

Der Namensraumbezeichner enthält die Nummer der *Major Version*, also lediglich die erste Stelle der vollständigen Versionsnummer. Die aktuelle Version hat den Namensraum-Bezeichner:

<http://reference.e-government.gv.at/namespace/xml-baukasten/1#>

Neue Komponenten, die im Rahmen einer *Minor Version* hinzukommen, müssen zum bestehenden Zielnamensraum hinzugefügt werden. Eine *Major Version* zieht eine Änderung des Namensraumbezeichners und damit zwangsläufig eine Inkompatibilität zur Vorgängerversion nach sich.

Im Umgang mit Schemakomponenten höherer/tieferer Versionen, die von einem Server oder Client nicht verstanden werden (inkompatibel zur unterstützten Version sind), gilt das *Must-Ignore-Prinzip*: XML-Elemente und Attribute in einem gültigen Dokument, die nicht verstanden werden, müssen ignoriert werden. Die Verarbeitung wird dabei nicht durch einen Fehler abgebrochen. Dieses Prinzip ist wohlbekannt von HTTP 1.1 (s. **[http-1.1]**) und wurde für XML publiziert in **[fxpp]**.

## (4) Verwendung des Basistypenschemas

Die folgende Anleitung zeigt, wie definierte Basistypen in einem neu zu spezifizierenden Schema eingebunden werden können.

### (4.1) Import des Basistypenschemas

Das Basistypenschema ist über das `xsd:import`-Element einzubinden. Dabei sind der Zielnamensraum des Basistypenschemas (Attribut `namespace`) und der Ort des Basistypenschemas (Attribut `schemaLocation`) anzugeben:

```
<xsd:import
  namespace=http://reference.e-government.gv.at/namespace/xml-baukasten/1#
  schemaLocation="eingebunden/xml-baukasten.xsd"/>
```

### (4.2) Referenzierung eines Elementtyps

Ein global deklarierter Elementtyp des Basisschemas wird durch Referenzierung über das `element`-Element von XML Schema (WXS) in ein Inhaltsmodell eingebunden:

```
<xsd:element ref="baukasten:Verfahren"/>
```

### (4.3) Referenzierung eines komplexen Typs

Eine weitere Möglichkeit der Einbindung von Basistypen besteht in der Deklaration eigener Elementtypen und Zuweisung von Typen des Basisschemas zu diesem Elementtyp. Dies erfolgt über das `type`-Attribut, dessen Wert einen global definierten Typ im Basistypenschema referenziert.

```
<xsd:element name="BeiLagen" type="baukasten:BeiLagenType"/>
```

## (5) Richtlinien für Basis-Komponenten

Durch die Verwendung des Schemas xml-eingangsprotokoll soll sichergestellt werden, dass die Gemeinsamkeiten eines Eingangsprotokolls gleich implementiert werden und eine Austauschbarkeit möglich wird.

Für die konkreten Anwendungsfälle eines elektronischen Eingangs mit anschließender Protokollierung wird empfohlen, jeweils ein spezifisches Schema pro Verfahren zu definieren.

Dafür soll der XML-Baukasten für die Erstellung der spezifischen Schemata, unter Ausformung des Elements `EingangsDaten` im Schema xml-eingangsprotokoll verwendet werden.

Die Struktur für elektronische Anträge wurde ursprünglich im Dokument [xml-a] festgelegt. Aufgrund der erfolgten Entwicklungen im Bereich des Eingangsprotokolls und des XML-Baukastens werden die dort festgelegten Strukturen in diese Empfehlung überführt. Das Dokument [xml-baukasten] ersetzt somit das Dokument [xml-a].

Die Empfehlungen sollen sinngemäß bei jeder Art von elektronischem Eingang verwendet werden, beispielsweise auch für EDIAKT Transaktionen oder sonstige Eingaben an eine öffentliche Stelle.

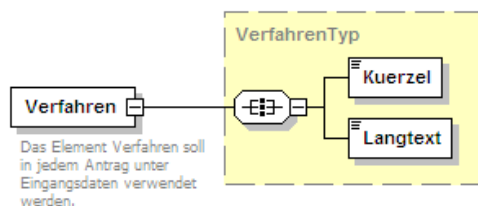
Weiterhin gültig bleiben die Empfehlungen aus dem Dokument [xml-g], wo die Modellierungsempfehlungen für XML-Elemente beschrieben werden.

### (5.1) Empfohlene Elemente

Es wird empfohlen für Anträge die folgenden Elemente aus dem XML-Baukasten unter dem Element `Eingangsdaten` zu verwenden:

#### 5.1.1. Verfahren

Das Element `Verfahren` soll in jedem Antrag unter `Eingangsdaten` verwendet werden.



#### 5.1.2. Weitere Elemente

Bei Bedarf werden hier weitere Elemente eingefügt.

### (5.2) Empfohlene Typen

Bei Benennung der weiteren Elemente unter dem Element `Eingangsdaten` wird empfohlen, diese entsprechend Ihrer Eigenschaft bzw. Rolle (siehe auch [xml-g]) zu benennen. Wenn es bereits einen entsprechenden Typ im XML-Baukasten gibt ist dieser dafür einzubinden.



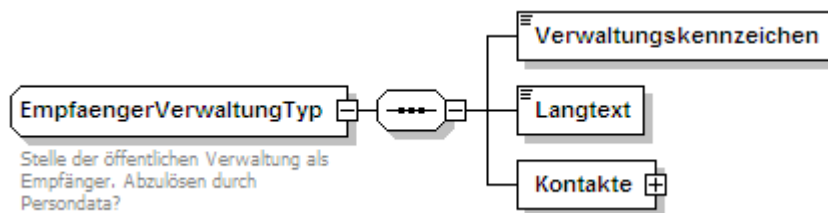
Rollen wie z.B. Absender oder Hauptwohnsitz sollten über die Typen der Person-Data-Spezifikation (**[persondata2]**) definiert werden.

Dadurch ergibt sich folgende Reihenfolge in der die Typen entnommen werden:

1. XML-Baukasten
2. Person-Data
3. Eigene Typen

Folgende Typen sind zur Zeit im XML-Baukasten definiert:

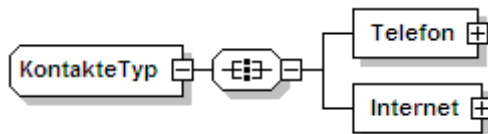
### 5.2.1. Empfänger



Wenn der Empfänger eine öffentliche Stelle der Verwaltung ist wird dieser Typ empfohlen.

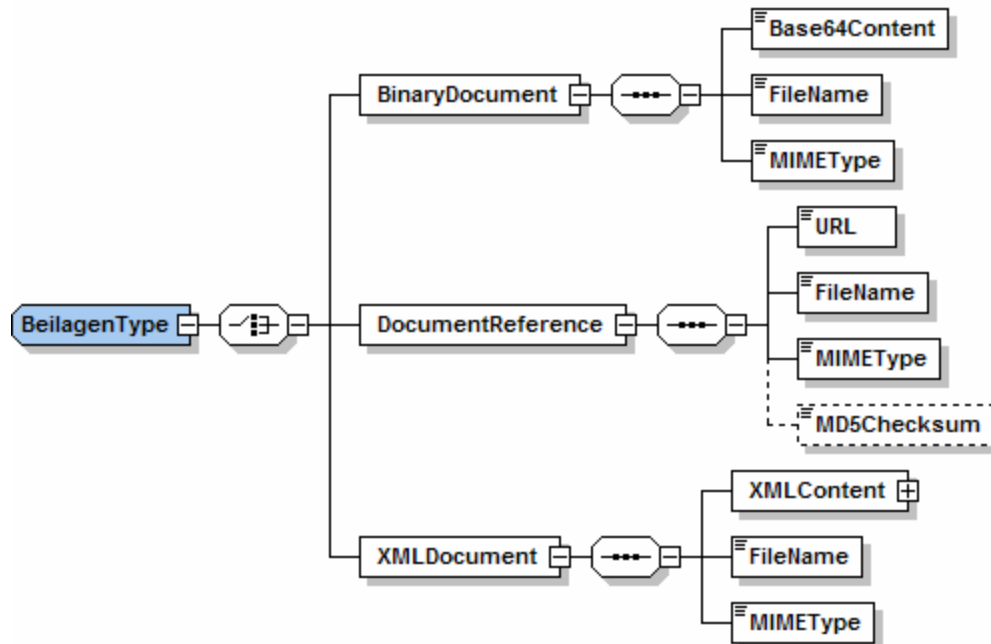
### 5.2.2. Kontakte

Der KontakteTyp besteht aus den beiden Elementen Telefon und Internet. Unter dem Element `pers:Typ` wird bei dem Element `Telefon` entweder „Telefon1“, „Telefon2“ und „Fax“ verwendet. Unter dem Element `Internet` wird die E-Mail-Adresse eingefügt.



### 5.2.3. Beilagen

Als allgemeine Basiskomponente für Beilagen wird hier der BeilagenType definiert. Der PayloadType ist aus dem derzeit gültigen EDIAKT-Schema entnommen.



## (6) Referenzen

### [fxpp]

Goland, Yaron Y; Schlimmer, Jeffrey C.. 2000. "Flexible XML Processing Profile (FXPP)". UPnP Forum Technical Committee.  
Internet: <http://www.upnp.org/download/draft-goland-fxpp-01.txt>.

### [http-1.0]

Berners-Lee, T.; Fielding; R.; Frystyk, H. 1996. "Hypertext Transfer Protocol -- HTTP 1.0". RFC 1945. Internet: <http://www.ietf.org/rfc/rfc1945.txt>.

### [http-1.1]

Fielding R. [u.a.]. 1999. "Hypertext Transfer Protocol -- HTTP/1.1". RFC 2616.  
Internet: <http://www.ietf.org/rfc/rfc2616.txt>.

### [nsp-1.0]

Bray, Tim; Hollander, Dave; Layman, Andrew. 1999. "Namespaces in XML". W3C Recommendation.  
Internet: <http://www.w3.org/TR/1999/REC-xml-names-19990114/>.

### [persondata2]

Naber, Larissa. 2004. "PersonData Struktur: XML Spezifikation". AG Kommunikationsarchitektur. Internet: <http://reference.e-government.gv.at/>.

### [rm]

Hörbe, Rainer. 2004. Release-Management für Anwendungen: Best Practice. AG Kommunikationsarchitektur.  
Internet:  
[https://w4.wien.gv.at/Groups/externe/e-government/kommarch/ergebnisdokumente/release\\_management/index.html](https://w4.wien.gv.at/Groups/externe/e-government/kommarch/ergebnisdokumente/release_management/index.html).

### [soap-1.1]

Box, Don [u.a.]. 2000. "Simple Object Access Protocol (SOAP) 1.1". W3C Note.  
Internet: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>.

### [soap-faults]

Liehmman, Michael. 2005. SOAP-Faults und deren Behandlung im Rahmen der Kommunikationsarchitektur des Bundes.

### [spec-szr]

Hörbe, Rainer. 2004. "Spezifikation Stammzahlen-Register (SZR-N)". BMI.  
Internet: <https://portal.bmi.gv.at/ref/szr/specszr.pdf>.

### [uri]

Berners-Lee, T. 1998. "Uniform Resource Identifiers (URI): Generic Syntax". RFC 2396. Internet: <http://www.ietf.org/rfc/rfc2396.txt>.

### [url]

Berners-Lee, T.; Masinter, L.; McCahill, M. 1994. "Uniform Resource Locators (URL)". RFC 1738. Internet: <http://www.ietf.org/rfc/rfc1738.txt>.

### [wsdl-1.1]

Christensen, Erik [u.a.]. 2001. "Web Services Description Language 1.1". W3C Note. Internet: <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>.

**[wsi-1]**

Ballinger, Keith [u.a.]. 2004. "Basic Profile Version 1.0". Web Services Interoperability Organization.

Internet: <http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html>.

**[wxs-1]**

Thompson, Henry S. [u.a.]. 2004. "XML Schema Part 1: Structures Second Edition". W3C Recommendation.

Internet: <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>.

**[wxs-2]**

Biron, Paul V.; Malhotra Ashok. 2001. "XML Schema Part 2: Datatypes". W3C Recommendation.

Internet: <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>.

**[xml-g]**

Luef, G.; Grandits, Franz. 2004. "E-Government: XML-Strukturen für Geschäftsobjekte". AG Kommunikationsarchitektur.

Internet: <http://reference.e-government.gv.at/>.

**[xml-a]**

Luef, G. 2004. "E-Government: XML-Strukturen für Antragsdaten". AG Kommunikationsarchitektur.

Internet: <http://reference.e-government.gv.at/>.