



ATIS-I-0000002

ATIS Standard on -

## ATIS XML SCHEMA DEVELOPMENT GUIDELINES



ATIS is the leading technical planning and standards development organization committed to the rapid development of global, market-driven standards for the information, entertainment and communications industry. More than 200 companies actively formulate standards in ATIS' Committees, covering issues including: IPTV, Cloud Services, Energy Efficiency, IP-Based and Wireless Technologies, Quality of Service, Billing and Operational Support, Emergency Services, Architectural Platforms and Emerging Networks. In addition, numerous Incubators, Focus and Exploratory Groups address evolving industry priorities including Smart Grid, Machine-to-Machine, Networked Car, IP Downloadable Security, Policy Management and Network Optimization.

ATIS is the North American Organizational Partner for the 3rd Generation Partnership Project (3GPP), a member and major U.S. contributor to the International Telecommunication Union (ITU) Radio and Telecommunications' Sectors, and a member of the Inter-American Telecommunication Commission (CITEL). ATIS is accredited by the American National Standards Institute (ANSI). For more information, please visit < <http://www.atis.org> >.

---

### **Notice of Disclaimer & Limitation of Liability**

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. ATIS SHALL NOT BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY ATIS FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL ATIS BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. ATIS EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

NOTE - The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to whether use of an invention covered by patent rights will be required, and if any such use is required no position is taken regarding the validity of this claim or any patent rights in connection therewith.
--

---

## **ATIS-I-0000002, ATIS XML Schema Development Guidelines**

*Published by*

**Alliance for Telecommunications Industry Solutions  
1200 G Street, NW, Suite 500  
Washington, DC 20005**

Copyright © 2011 by Alliance for Telecommunications Industry Solutions  
All rights reserved.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher. For information contact ATIS at 202.628.6380. ATIS is online at < <http://www.atis.org> >.

Printed in the United States of America.

# ATIS XML SCHEMA DEVELOPMENT GUIDELINES

**Alliance for Telecommunications Industry Solutions**

Approved July 1, 2011

## **Abstract**

This document provides ATIS guidelines for the development of XML schema.

## FOREWORD

---

The Alliance for Telecommunication Industry Solutions (ATIS) serves the public through improved understanding between carriers, customers, and manufacturers. The mandatory requirements are designated by the word *shall* and recommendations by the word *should*. Where both a mandatory requirement and a recommendation are specified for the same criterion, the recommendation represents a goal currently identifiable as having distinct compatibility or performance advantages. The word *may* denotes an optional capability that could augment the standard. The standard is fully functional without the incorporation of this optional capability.

Suggestions for improvement of this document are welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, 1200 G Street NW, Suite 500, Washington, DC 20005.

The XML harmonization Team was responsible for the development of this document.

## TABLE OF CONTENTS

---

<b>1 BACKGROUND, OBJECTIVE, AUDIENCE, &amp; SCOPE .....</b>	<b>1</b>
1.1 BACKGROUND .....	1
1.2 OBJECTIVE .....	1
1.3 AUDIENCE.....	1
1.4 SCOPE.....	1
1.5 GUIDELINES GOVERNANCE .....	1
<b>2 REFERENCES.....</b>	<b>1</b>
2.1 NORMATIVE REFERENCES .....	1
2.2 INFORMATIVE REFERENCES.....	2
<b>2 DEFINITIONS, ACRONYMS, &amp; ABBREVIATIONS.....</b>	<b>3</b>
2.1 ACRONYMS & ABBREVIATIONS .....	3
2.2 DEFINITIONS .....	3
<b>3 SCHEMA DEVELOPMENT - INFORMATIVE.....</b>	<b>3</b>
3.1 DATA MODEL USING XML SCHEMA.....	3
3.1.1 <i>Schema Specification Version</i> .....	4
3.2 XML SCHEMA DESIGN CONSIDERATIONS.....	4
3.2.1 <i>Developing a Single Schema or a Collection of Related Schemas</i> .....	4
3.2.2 <i>Use of Data Type Constraints</i> .....	5
3.2.3 <i>Schema Design Patterns</i> .....	5
3.2.4 <i>Designing for Resiliency</i> .....	5
<b>4 RECOMMENDATIONS FOR SCHEMA DEVELOPERS - NORMATIVE.....</b>	<b>6</b>
4.1 XML SCHEMA DESIGN REQUIREMENTS .....	6
4.2 XML SCHEMA DESIGN RECOMMENDATIONS .....	6
4.2.1 <i>XML Schema Constructs to Avoid</i> .....	7
4.3 XML SCHEMA CODING RECOMMENDATIONS .....	8
<b>5 SCHEMA MANAGEMENT - NORMATIVE .....</b>	<b>9</b>
5.1 FORWARD AND BACKWARD COMPATIBILITY AND SCHEMA VERSIONING .....	9
5.1.1 <i>Definitions</i> .....	9
5.1.2 <i>Schema Update Guidelines for Forward Compatibility</i> .....	10
5.1.3 <i>Schema Update Guidelines for Backward Compatibility</i> .....	10
5.2 GUIDELINES FOR SCHEMA EXTENSIONS.....	11
5.3 SYNCHRONIZATION OF STANDARD DOCUMENT AND XML SCHEMA FILE .....	12
5.3.1 <i>XML Schema Filename Naming</i> .....	12
5.3.2 <i>Cross-Referencing Standard Document and XML Schema file(s)</i> .....	12
<b>6 ONLINE PUBLICATION OF THE SCHEMA - NORMATIVE .....</b>	<b>13</b>
<b>7 LEGAL NOTICES - NORMATIVE .....</b>	<b>13</b>

---

## TABLE OF FIGURES

---

FIGURE 1: VERSION COMPATIBILITY .....	10
---------------------------------------	----

# ATIS XML Schema Development Guidelines

## 1 BACKGROUND, OBJECTIVE, AUDIENCE, & SCOPE

---

### 1.1 Background

It was realized that common guidelines were needed for the development of XML schemas by all ATIS Committees and Forums. This document is the result of these guidelines developments.

### 1.2 Objective

This document provides a set of guidelines that will help unify the style and maximize the readability, reusability, and the forward and backward compatibility of XML schemas developed in the various ATIS Committees. This document provides recommendations that will help members avoid common pitfalls and choose consistent naming conventions and styles when developing schemas.

### 1.3 Audience

This document is expected to also be useful to other architects and software engineers who develop XML schemas for all ATIS Committees. It is assumed that the reader has a working knowledge of W3C XML Schema and the XML language.

### 1.4 Scope

The document covers XML schema design, coding, and issues regarding evolution and compatibility of successive schema versions. The recommendations within this document are the result of a collaboration of ATIS Committees and using inputs from other accredited standards organizations.

### 1.5 Guidelines Governance

It is up to each ATIS Committee to practice these Guidelines. Each Committee should keep a record of the following:

- ◆ Deviations from these Guidelines, along with rationales.
- ◆ Any additional Guidelines adopted by the Committee.

Any changes to the record should be reported to ATIS Standards Development staff.

## 2 REFERENCES

---

### 2.1 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this ATIS Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this ATIS Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

- [1] W3C XML Schema Part 0: Primer, 2004<sup>1</sup>.
- [2] W3C XML Schema Part 1: Structures, 2004.<sup>2</sup>
- [3] xFront, *Multi-Schema Project: Zero, One, or Many Namespaces?*<sup>3</sup>
- [4] Oracle Sun Developer Network, *Introducing the Design Patterns in XML Schemas*, November 9, 2006.<sup>4</sup>
- [5] ATIS-I-000001, *ATIS Namespace Format*, 2011.<sup>5</sup>

## 2.2 Informative References

- [6] ITU M.3030, *Telecommunications Markup Language (tML) Framework Series M: TMN and Network Maintenance: International Transmission Systems, Telephone Circuits, Telegraphy, Facsimile and Leased Circuits Telecommunications Management Network*, 2002.<sup>6</sup>
- [7] xFront, *Global versus Local*.<sup>7</sup>
- [8] U.S. EPA, *XML Design Rules for the Environmental Information Exchange Network*, 2003.<sup>8</sup>
- [9] W3C *Extensible Markup Language (XML) 1.0*, 2008.<sup>9</sup>
- [10] W3C *Namespaces in XML*, 2009.<sup>10</sup>
- [11] W3C *XML Schema Definition Language (XSD) 1.1 Part 1: Structures*, 2009.<sup>11</sup>
- [12] W3C *XML Schema*, 2005.<sup>12</sup>
- [13] W3C *XML Schema Versioning Namespace*, 2007.<sup>13</sup>

---

<sup>1</sup> This document is available at < <http://www.w3.org/TR/xmlschema-0/> >.

<sup>2</sup> This document is available at < <http://www.w3.org/TR/xmlschema-1/> >.

<sup>3</sup> This document is available at < <http://www.xfront.com/ZeroOneOrManyNamespaces.html> >.

<sup>4</sup> This article is available at < [http://developers.sun.com/jenterprise/archive/nb\\_enterprise\\_pack/reference/techart/design\\_patterns.html](http://developers.sun.com/jenterprise/archive/nb_enterprise_pack/reference/techart/design_patterns.html) >.

<sup>5</sup> This document is available from the Alliance for Telecommunications Industry Solutions (ATIS), 1200 G Street N.W., Suite 500, Washington, DC 20005. < <https://www.atis.org/docstore/default.aspx> >

<sup>6</sup> This document is available from the International Telecommunications Union. < <http://www.itu.int/ITU-T/> >

<sup>7</sup> This document is available at < <http://www.xfront.com/GlobalVersusLocal.html> >.

<sup>8</sup> This document is available from the U.S. Environmental Protection Agency (EPA). < <http://xml.gov/documents/completed/epa/EPAGuideSec1.pdf> >

<sup>9</sup> This document is available at < <http://www.w3.org/TR/REC-xml> >.

<sup>10</sup> This document is available at < <http://www.w3.org/TR/REC-xml-names/> >.

<sup>11</sup> This document is available at < <http://www.w3.org/TR/xmlschema11-1/> >.

<sup>12</sup> This document is available at < <http://www.w3.org/2001/XMLSchema> >.

<sup>13</sup> This document is available at < <http://www.w3.org/2007/XMLSchema-versioning> >.

## 2 DEFINITIONS, ACRONYMS, & ABBREVIATIONS

---

### 2.1 Acronyms & Abbreviations

ATIS	Alliance for Telecommunications Industry Solutions
DTD	Document Type Definition
HTTP	HyperText Transfer Protocol
QoS	Quality of Service
tML	telecommunications Markup Language
W3C	World Wide Web Consortium
URL	Uniform Resource Locator
XML	eXtensible Markup Language
XSD	XML Schema Definition

### 2.2 Definitions

<b>Camel Case</b>	The practice of writing compound words together in which words are joined together without spaces, with the first letter capitalized. In UpperCamelCase, the first letter of the first word is capitalized, whereas in the lowerCamelCase the first letter of the first word is in lower case.
<b>Non-deterministic content model</b>	A schema is considered to have a non-deterministic content model if a XML parser is unable to clearly determine the structure to validate with the schema. When validation is attempted on a non-deterministic content model, a parser may generate an error.
<b>Schema validator</b>	A schema validator is a tool or program which can parse a schema file to check if it is well-formed and also validate it.
<b>XML Application</b>	An XML Application is a specific XML vocabulary that contains particular elements and attributes. It is not a software program that uses XML. XML applications limit the very flexible rules of XML to a finite set of elements of certain types by the definition of an XML Schema. An XML Application is defined by one or more namespaces.
<b>XML Document</b>	An XML document written in accordance with the W3C XML 1.0 specification. An XML Document is well-formed or it is not XML. It does not necessarily have to be valid to be well-formed
<b>XML Instance Document</b>	A document or message written in XML which is well-formed and valid with regard to some XML Schema. An XML document that satisfies all the rules of some XML Application.
<b>XML Schema</b>	Schemas can be written in a variety of languages, but in this document, an XML Schema is one written according to W3C XML Schema standards.

## 3 SCHEMA DEVELOPMENT - INFORMATIVE

---

### 3.1 Data Model using XML Schema

XML allows arbitrary data definitions using ad-hoc tags. The XML document becomes practical for use when it is constrained by a well-defined structure. An XML schema provides a means to define rules, semantics, and structure for XML documents. A schema provides programmatic validation of a

structured XML document. An XML schema is defined using a XML format in a file with an “.xsd” extension.

XML Schema was approved as a W3C Recommendation on 2 May 2001 and a second edition was published on 28 October 2004. As of the writing of this document, the published version of XML Schema is 1.0. W3C XML Schema WG is working towards completion of XML Schema 1.1, which is intended to be mostly backward compatible with XML Schema 1.0.

### 3.1.1 Schema Specification Version

The namespace for XML Schema 1.1 [12] is the same as XML Schema for 1.0 document and the namespace for XML instance document remains as “http://www.w3.org/2001/XMLSchema-instance”. This allows schema developers to develop XML schema 1.0 documents without worrying about updating namespace when adding 1.1 features. XML Schema 1.1 introduces a new namespace for version control [13].

An XML document conforming to a 1.0 schema can be validated using a 1.1 schema validator, but an XML document conforming to a 1.1 schema will likely not validate using a 1.0 schema validator. Schema developers will have to use appropriate discretion when deciding to use version 1.1 constructs in XML schema definitions.

## 3.2 XML Schema Design Considerations

### 3.2.1 Developing a Single Schema or a Collection of Related Schemas

The schema definition language allows constructs to import schema from other documents. When developing a set of related schemas, namespace considerations become important in how the resulting XML instance document references other schemas.

- ◆ Reference using <import>: The import element allows references to schema components from schema documents with different target namespaces. This is the most common schema design approach and is also referred to as *heterogeneous namespace design* [3].
- ◆ Reference using <include>: The include element adds the schema components from other schema documents that have the same target namespace (or no specified target namespace) to the containing schema. The include element thus allows you to add all the components of an included schema to the containing schema. This gives rise to two different design approaches [3]:
  - *Homogeneous namespace design* approach: In this design approach, all related schemas being developed are assigned the same target namespace.
  - *Chameleon namespace design* approach: In chameleon namespace design approach, there is one or more supporting schema defined with no target namespace, the main schema includes the supporting schema(s). The supporting schema(s) take the namespace of the main schema.

The homogeneous or the chameleon design approach allows ability to redefine a type, group, and attribute group definitions defined in a supporting schema. Type redefinition affects the elements in the including schema as well as those in the included schema. Thus redefined types can interact with derived types and generate conflicts.

It is left to the ATIS Committees to choose the strategy that best meets their specific needs.

### 3.2.2 Use of Data Type Constraints

XML schema constraints are applied to data type definitions in order to perform XML instance document validation. When developing schemas, it is possible to provide a complete and rigorous set of constraints for each data element (tightly constrained) or to take a minimalistic approach (loosely constrained) in having the schema enforce data validation rules. The decision on how much to constrain depends upon the requirements of the issue being addressed. It is left to the ATIS Committees to choose which direction to proceed on an issue-by-issue basis.

### 3.2.3 Schema Design Patterns

The following four structural design patterns are commonly mentioned related to XML schema design [4]:

- ◆ *Russian Doll*: The Russian Doll design provides a single global element in the schema file. All child elements are defined within this single element definition hierarchy. The design does not promote element reusability and requires definition of elements in a single schema file.
- ◆ *Salami Slice*: In this design approach, multiple root elements are defined but no complexType is defined. The design supports reusability of individual elements.
- ◆ *Garden of Eden*: A schema design approach in which all elements and types are exposed globally and complex types are defined through reference to reusable global elements. The characteristic of this design pattern is that there are many possible root elements.
- ◆ *Venetian Blinds*: A schema design approach in which types are created first from which elements are built. The simpleTypes and complexTypes are created so as to maximize reuse. The characteristic of this design approach is that there is a single root element.

### 3.2.4 Designing for Resiliency

When developing a schema, one of the goals is to make it resilient to changes and provide flexibility that anticipates future needs of the schema users. This section discusses some of the mechanisms that are available for providing flexibility and extensibility in an XML schema model.

#### 3.2.4.1 Using Wildcards

The W3C schema allows constructs such as <any> and <anyAttribute> to allow instance document to contain XML data not directly constrained by the content model of the defining XML Schema. This allows extensibility mechanism with some degree of control that can be specified using “namespace” and “processContents” attributes (see XML Schema Primer [1] and W3C XML Schema Part 1 [2]). The namespace attribute, for example, can be used to constrain the XML data to a set of predefined namespaces thus providing some data validation that can be performed on these extended XML data within an instance document. The “processContents” attribute controls how this extended XML data is validated by the XML validator.

The <any> wildcard can allow vendors to develop vendor specific functionalities not defined in the defining schema. A careful use of <any> and <anyAttribute> can help build a schema content model more resilient to change and less prone to schema churn. However on a note of caution, with <any> it is possible to inadvertently allow creation of non-deterministic content models which can cause issues with some XML parsers. A schema developer will have to consider this constraint when deciding to use <any>.

### 3.2.4.2 Using substitutionGroup

The content model of substitution group members is related to each other by type derivation. The replaceable element is called the head element and has to be defined in the schema's global scope. In essence, the substitutionGroup construct helps build a collection of elements that can be specified using a generic element.

The substitutionGroup construct can be helpful in following cases:

- ◆ *Development of related class hierarchies:* Creating class hierarchies can enable object-oriented programming languages to take advantages of such inheritance.
- ◆ *Customizing an external schema for a specific need:* If an element in an imported schema is defined globally, it is possible to create a substitutionGroup for this element through construction of a derived type and allowing substitution of this derived type in a complex data structure.

Class hierarchies and substitutionGroup results in tight coupling between data structures and can lead to brittle and inflexible design. Constructs such as <choice> can be used instead to create composite content model. The composite design can lead to simplicity and a decoupled design.

## 4 RECOMMENDATIONS FOR SCHEMA DEVELOPERS - NORMATIVE

---

### 4.1 XML Schema Design Requirements

The author *shall*:

1. Use the ATIS namespace format and guidelines as defined in ATIS-I-0000001 [5].

### 4.2 XML Schema Design Recommendations

The suggestions below are based on the combined "lessons learned" of multiple ATIS Committees. They may or may not apply to one specific design pattern and are offered as a potential aid to schema developers.

The authoring committee *should*:

1. Discuss and adopt an XML schema design pattern that best satisfies their specific needs. See section 4.2.3 for some commonly-considered XML schema design patterns.
2. Adhere to a change-control process that links the published schema(s) to the associated ATIS Standard(s).

The author *should*:

3. Declare all Simple and Complex Types in the global scope of the current schema.
4. All elements and attributes are declared within a root element. The root element encapsulates all elements and attributes.
5. Provide a two-part schema version number, consisting of a major version number and a minor version number.
  - o *Major version number*: A positive integer, which should start at 1 and be incremented by 1 whenever a non-forward compatible change is made to the schema.
  - o *Minor version number*: A non-negative integer, which should start at 0, be incremented by 1 whenever a forward compatible change is made to the schema, and be reset to 0 whenever the major version number is incremented.
6. Not include the minor version number of the schema as part of the namespace.
7. Ensure the <schema> element of the schema contains the “version” attribute, with value equal to the full version number of the schema in <major>.<minor> format. The initial value is expected to be “1.0”.
8. Ensure all schemas have at least two namespaces: the W3C XML Schema namespace and the namespace related to the companion ATIS Standard.

Example schema attributes for version 1.3 of the ATIS IPTV Interoperability Forum (IIF) Emergency Alert System schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.atis.org/ServiceDelivery/IPTV/EmergencyAlert/1/schema"
  xmlns:eas="http://www.atis.org/ServiceDelivery/IPTV/EmergencyAlert/1/schema"
  version="1.3">
  . . . . .
</schema>
```

The author *should* consider the following guidelines when using “id” attribute in the schema:

9. Use the “id” attribute of the schema to identify the schema. The “id” attribute is an optional attribute within the schema element. Defining an “id” can allow better debugging support for programmatic identification of XML elements references to schema. For example, the ATIS document number could be used for the “id” attribute.

#### 4.2.1 XML Schema Constructs to Avoid

The author *should* consider the following guidelines for XML Schema constructs:

1. Do not use <all> (use <sequence> or <choice>). The <sequence> and <choice> forces a fixed ordering of child elements in the instance document.
2. Do not use processing instructions. Processing instructions do not form part of the document but are passed to the application to perform application specific functions. Processing instructions do not have to follow internal structure and as such have little use in schema definitions.
3. Do not use notations. The notation element is used to describe format of non-XML data within XML document. They are not used in validation but are used to validate strings as members of NOTATION simple type. They can be used to pass processing instructions to applications.

4. Do not use abstract types. Use of abstract type allows extension and consequently possibility of vendor specific extensions of the abstract data type in derived schema. These can potentially cause interoperability issue and incompatibilities.
5. Do not use Document Type Definition (DTD) or XML style comments. The <annotation> and <documentation> elements provide construct for comments and information. The XML style comments are not useful for processing of an instance document.
6. Do not use XML groups or group redefinition. The redefinition of a group can generate conflict between processing of redefined type and derived type instance data.
7. Do not use substitution groups. The substitution group construct creates tight coupling and adds complexity, which can lead to brittle and inflexible design.
8. Do not use default/fixed values. Including such attribute uses will tend to mislead readers of the schema document, because the attribute uses would have no effect.

#### 4.3 XML Schema Coding Recommendations

The author *should* utilize the following guidelines for type, element, and attribute names:

1. Utilize common industry or business names where possible to ensure consistency between industry or business documentation and schemas.
2. Avoid abbreviations; make names descriptive.
3. Use the same data type for elements or attributes having the same name.
4. Limit tag names to 20 characters (not including prefix and ':').
5. Use alpha-numeric characters only.
6. If acronyms are used, they *shall* be defined in the accompanying standard document. Using an acronym, the capitalization rules of the acronym remain, for example Quality of Service may use the acronym QoS.
7. Avoid period and dash characters.
8. Use UpperCamelCase - e.g., "PersonName" for defining element names.
9. Use lowerCamelCase for defining attribute names.
10. Use lowerCamelCase for defining simpleType names and UpperCamelCase for defining complexType names.
11. End data type names with "Type".
12. Use singular names unless the concept itself is plural.
13. Use names that only contain nouns, verbs, prepositions, and adjectives.

The author *should* consider element as the default model type and use the following guidelines:

14. If the item can be considered an independent object, make it an element.
15. If the item needs to be re-usable, make it a global type.
16. If the item does not have any children and describes a characteristic of an element and is never multivalued, make it an attribute.

The author *should* utilize the following general guidelines:

17. Declare elements as optional unless absolutely required.
18. Use minOccurs="0" instead of nillable="true".

19. Avoid defining attributes with their default values (for example, minOccurs=1, maxOccurs=1).
20. Define simpleTypes whenever possible to clarify context.
21. Create a global type when the element is to be reusable. Global types are defined directly under the schema element.
22. All types are to be defined globally.
23. Use elementFormDefault = "qualified".
24. Use attributeFormDefault = "unqualified".
25. Use UTF-8 character encoding: <?xml version="1.0" encoding="UTF-8"?>.
26. The <documentation> elements are to be used wherever re-use is likely and more clarity is desired.
27. The lang attribute is to be used to specify language.
28. Use <annotation> to describe all data type definitions; minimally include the data type name.
29. Use only dateTime data type for date and time items.
30. Use only <sequence> or <choice> where a compositor is required.

The author *should* consider the following guidelines when adding constraints:

31. Identify facets that constrain the range of values.
32. Choose the appropriate simple type. For example, use of the nonNegativeInteger or positiveInteger simple type is preferred to integer.
33. When utilizing the string type, if possible use "maxLength" facet.
34. Patterns may be used to provide additional restrictions. For example, when defining a 15 digit international phone number, the pattern can be (ITU-T Recommendation E.164):
 

```
<restriction base="string">
  <pattern value="([1-9][0-9]{0,2})(-[1-9][0-9]{0,4})?(-[1-9][0-9]{0,13})(-[0-9]+)?"/>
</restriction>
```
35. When a simple type takes only a predefined set of values, consider using the enumeration facet to restrict the legal values of the simple type.
36. When the data type can contain two or more sets of values that can independently be constrained, consider using a <union>.

## 5 SCHEMA MANAGEMENT - NORMATIVE

---

### 5.1 Forward and Backward Compatibility and Schema Versioning

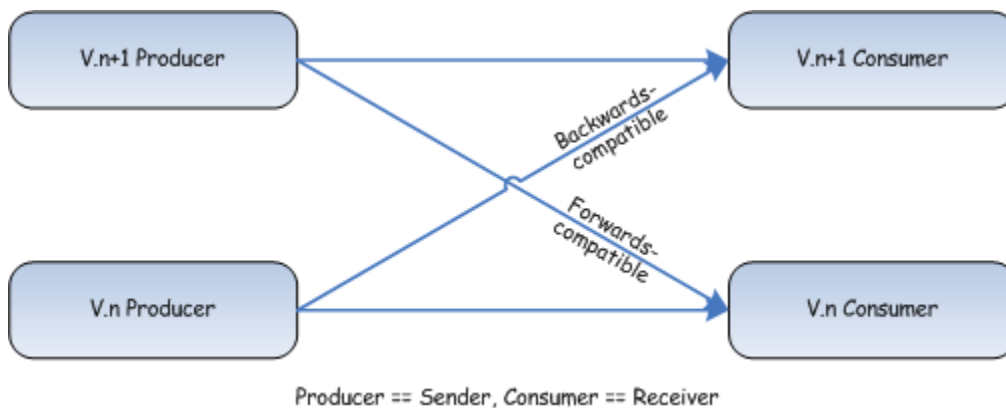
An XML Producer sends an XML Instance Document based on some schema vocabulary and version to an XML Consumer. This section describes best practices to avoid issues that could arise when the Producers and Consumers have different versions of the schema vocabulary.

#### 5.1.1 Definitions

- > *Forward compatibility* means that a newer version of a producer can be rolled out in a way that does not break existing consumers.
  - ◆ Of course the older consumer will not implement any new behavior, but a producer can send a newer version of an XML instance document and still have the instance

successfully processed.

- > *Backward compatibility* means that a newer version of a consumer can be rolled out in a way that it is not broken by existing producers.
  - ◆ A producer can send an older version of a message to a consumer that understands the new version and still have the message successfully processed.



**Figure 1: Version Compatibility**

### 5.1.2 Schema Update Guidelines for Forward Compatibility

The following guidelines are provided to ensure forward compatibility, if forward compatibility is required:

1. Assume XML instance parsers follow the “Must Ignore” rule. This rule means that a consumer will parse a received XML Instance Document even if it fails to validate against the known schema (based on the documents namespaces). In this case, anything that is not understood by the parser is silently ignored.
2. Extend content in original namespace because XML Schemas with different namespace names are incompatible.
3. Include a version attribute in the schema element to represent a minor version number. XML Schemas with same namespace names but different version attribute values are compatible (if and only if “Must Ignore” rule is followed).
4. If an element is added that has to be understood, update the namespace name with the new version number thereby forcing instance parsers to fail or update their reference schemas.
5. Avoid the use of <any> and <anyAttribute>, because they can lead to non-deterministic content model that can result in parsing errors with some parsers.

### 5.1.3 Schema Update Guidelines for Backward Compatibility

The following guidelines are provided to ensure backward compatibility, if backward compatibility is required.

**Do:**

1. Add new data as optional elements or attributes (or enumerations). The descendants of these new elements need not be optional. Make new content optional with minOccurs="0" for elements and use="optional" for attributes.
2. Only increase the value of maxOccurs as required.
3. Utilize <choice> to replace an existing element with a new one. Introduce the new element and place it along with the existing element in a <choice> compositor.
4. Define simple and complex types explicitly as global types. Use named types, not anonymous types. This makes it easier to migrate a type by adding a new named type which is a modified version of the original. Both types are then included in the new schema.

**Don't:**

5. Change the meaning or semantics of existing components.
6. Add required components.
7. Remove required components.
8. Reduce the multiplicity (also known as cardinality or recurrence) of elements.
9. Use simple type restrictions other than maximum length.
  - NOTE – The maximum length value should only ever be increased.
10. Restrict a previous components content model (such as changing a choice to a sequence).
11. Use the XML mechanism "redefine" on any standard schema complex or simple types.
12. Alter the order of elements in a sequence or rearrange the schema element groupings or schema hierarchy. If the author inserts a new element into a sequence, do not reorder the elements that are already there.

## 5.2 Guidelines for Schema Extensions

It is understood that the schema authors cannot anticipate all future needs of a schema in advance. The author *should* utilize schema definition that provides flexibility and allow implementers to carry private or custom data. In general, schema authors *should* anticipate the need for supporting private data by including constructs to support inclusion of some data in a generic way (for example, name-value pair). However, such inclusion might not suffice a more complex need, where, for example, validation rules assertions and policies might be required.

The schema author *should* utilize one of two possible ways of extending support for private data in a schema:

1. Anticipate needs of extensions to specific portions of schema.
2. Type Inheritance using <extension>.

Use of such extension is discouraged as this potentially will lead to interpretability issues. For example, if an implementer extends a standard schema using <extension> to add an implementer specific elements and attributes, a resultant XML data might not parse correctly by other implementers who base their applications on the original standard schema.

The <extension> might also lead to backward compatibility issues.

### 5.3 Synchronization of Standard Document and XML Schema File

The XML schema(s) related to an ATIS Standard is realized in one or more XML schema files with extension “.xsd”. During the development of the standard, the .xsd schema file(s) *should* be zipped and stored with the companion contribution or working text. The .xsd file(s) *should* be considered the normative version of the XML schema. The .xsd file(s) *should* be well-formed and valid at all times. Any other XML schema files imported into the main XML schema file(s) *should* either be present in the same zip file or easily accessible so that complete validation of the normative XML schema is possible.

Schema updates *should* always be applied to the schema file first, validated, and then copied and pasted into the associated ATIS Standard to ensure that the XML text in the document is always identical to the XML schema file and valid.

#### 5.3.1 XML Schema Filename Naming

The XML schema file *should* be named such that it is obvious which ATIS Standard it is related to.

*Format during development of the ATIS Standard:*

Determined by each ATIS Committee or Forum.

*Format on publication of the ATIS Standard:*

`<ATIS_documentnumber>[document version][_multi-schema discriminator] .xsd`

Example:

`ATIS-0800013.v001.xsd`

If there are multiple schema files related to a single standard document, then the schema author *should* append distinguishing characters after the document version number (if present). The schema author *should* make the discriminators meaningful if possible.

Example:

`ATIS-0800013.v001_aaa.xsd`

`ATIS-0800013.v001_bbb.xsd`

`ATIS-0800013.v001_ccc.xsd`

Note that this file naming scheme means that the schema file name(s) *should* be changed whenever the published ATIS Standard is changed, even if the schema has not been modified.

#### 5.3.2 Cross-Referencing Standard Document and XML Schema file(s)

The following text *should* be included in the ATIS Standard Document and the XML Schema file respectively.

- ◆ Standard document reference to the .xsd file:

“The XML schema fragments contained in this document are an informative version of the definition of the XML schema defined in <XML schema filename> with schema attributes:

targetNamespace <schema targetNamespace name attribute> and version <schema version attribute>. If any parts of the XML text of this document contradict the normative part of the corresponding XML schema file, then the normative part *shall* be used as the definitive specification.”

NOTE – Include in the reference list for the document any schema files that are imported into the main schema since these are also normative schemas.

- ◆ XML schema file reference to the document:

```
<annotation>
```

```
<documentation xml:lang="en">
```

```
This XML document contains the normative definition of the XML schema defined in ATIS-xxxxxxx. In the event the informative part of the corresponding standard contradicts this XML schema, the XML schema in this file SHALL be used as the definitive specification.
```

```
</documentation>
```

```
</annotation>
```

This entry in the schema file allows a reader to access the correct version of the associated standard.

## 6 ONLINE PUBLICATION OF THE SCHEMA - NORMATIVE

---

It is not anticipated that the schemas will be published through an ATIS URL.

## 7 LEGAL NOTICES - NORMATIVE

---

All legal notices are to be inserted at the start of all XML schema files. The ATIS administrative staff is responsible for inserting all legal notices. The author of the XML schema *should not* insert such legal notices.