



# **OpenTravel Alliance Revised Namespace and Versioning Specification**

Version 1.0  
May 21, 2003

## ***Objectives***

Identify a set of Namespace & Versioning goals that satisfy both OTA's Best Practices and any new OTA agreed namespace and versioning requirements that also maps well with existing technology standards, guidelines and implementations.

The intent is to provide clear and consistent best practices that can help simplify OTA's task for managing XML content for the Travel Industry.

## ***Namespace and Versioning Goals***

- I. The versioning mechanism should provide an easy way to identify the two basic ways a XML Schema and instance doc changes:
  - A) Extensions to a Schema via adding new content which does not invalidate the previous version (e.g. minor version changes).
  - B) Structural content or data type changes where the previous content would not validate against the new schema (e.g. major version change).
- II. Multiple minor version messages of a particular base message Schema (or major version) will all validate against the latest base Schema version (e.g. forward compatibility: message versions '2.012', '2.037' and '2.500' all validate against Schema version '2.500').
- III. Common data types Schema files (e.g. type definitions only) are version independent of consuming messages and their content may transcend messages of multiple versions (of either major and minor message version).
- IV. Unique XML instance message versions must correlate to a message schema version without compromising forward compatibility of minor version messages.
- V. The Domain namespace for OTA XML content must support OTA versioning goals for Schemas and instance messages (e.g. a common schema file can validate multiple XML instance messages at the minor version level).
- VI. The Domain namespace for OTA XML content must support multiple schema file development without forcing use of any qualified namespace prefixes.
- VII. Maximize the reusability of OTA data type content.

## ***Best Practices for setting Namespace and Version***

- I. There will be one OTA domain targetNamespace for all XML message Schemas versions which is targetNamespace='http://www.opentravel.org/OTA/2003/05'.<sup>1</sup> However, the domain name will include an extension of alpha or beta corresponding to member review and public review, if additional releases are necessary, they would continue with gamma, delta, etc. This approach supports a consistent way to manage and identify OTA's XML based, transaction assets both internally and externally. It also avoids the need for using explicit prefixes in both schema and instance docs.

### ***Example:***

```
http://www.opentravel.org/OTA/2003/05 (publication "official" namespace)
or
http://www.opentravel.org/OTA/2003/05/alpha (member review namespace)
or
http://www.opentravel.org/OTA/2003/05/beta (public review namespace)

Usage:

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.opentravel.org/OTA/2003/05"
  xmlns="http://www.opentravel.org/OTA/2003/05"
  version="1.000"
  id="OTA2003A">
```

- II. There will be no namespace for any Common OTA data type .xsd Schema files. Common data types Schema files (i.e. type definitions only) are version independent from message schemas in which they are included. This permits their content to be included in multiple versions of a given request or response message.

### ***Example:***

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  version="4.123"
  id="OTA2003A2003B">
```

- III. Each XML instance document produced by the OTA Schemas should specify a default namespace and that should be the OTA namespace defined above. Also, a namespace prefix of "OTA" is to be reserved for the OTA namespace and used where OTA is required not to be a default namespace to satisfy unique business needs. This allows for a standard way for OTA namespace content to be merged with other Industry or Trading Partner namespace content.

---

<sup>1</sup> Starting in 2003, the year and month on this targetNamespace is used only for the initial publication of the OTA specification. This value does not change with the addition of new messages. If OTA were to deprecate the 2003 baseline specifications, then this value would change to support OTA's new baseline specifications (which should only occur on a 3 or 4 year cycle).

- IV. The 'version' attribute on the root tag of OTA message .xsd schema files will contain a decimal value of format n.nnn identifying both the major version base and a minor version value (e.g. version="1.127" where '1' is the major version base and '127' is the minor version value). This version number will be set by the OTA for each schema, based on the changes done to that message schema from one release to the next.
- **Major version change**: Structural content or data type changes where the previous content would not validate against the new schema.
  - **Minor version change**: Extensions to a Schema by adding new content that does not invalidate the previous version (i.e., does not break forward compatibility).

This means that messages that validate against an older base message Schema (or major version) will also validate against the latest base Schema version (e.g. a message that validates against schema version '2.012' will also validate against versions '2.037' and '2.050').

For 2003A, the version number will be set to 1.000 for all schemas, as this is the first release that uses this new versioning approach.

- V. The 'version' attribute on the root tag of all OTA Common data type .xsd schema files will contain independent self describing version values, using the same format as described in point IV above.
- VI. The root tag of XML payload messages will contain a 'version' attribute, the value of which will identify both the major version base and a minor version value of the schema against which it was designed.

**Example:**

```
XML instance value:
  version="1.050"

matches schema value:
  Version="1.050"
```

- VII. The 'id' attribute on the root tag of OTA message .xsd schema will contain the release and the 'id' attribute on the root tag of OTA Common data type .xsd schema will contain the range release. This will provide a practical way to identify and manage the full content of a specific OTA specification release.

**Example:**

```
Message schema files:
  id="OTA2003A"

CommonType schema files
  id="OTA2003A2003A"
```