

```
on_v3_0.xsd", xmlns:nei="http://www
elementFormDefault=
version="3.0">
<xsd:include schemaL
- <!--
KML 3.0 Start of Schema Header
-->
_ <xsd:annotation>
ngennetwork</xsd:documentation>
XML 3.0 Point data</xsd:documentation>
Available:http://
documentation>
<xsd:documentation
Environmental Protection input format</xsd:documentation>
encoding="UTF-8" ?
user</xsd:documentation>
ce="http://www.e
p://www.w3.org/1.0 encoding="UTF
p://www.epa.gov/ε
default="qualified" attribute namespace="http://
:"http://www.
chemaLocation="EN_NEI http://www.e
Default="qual
'>
chemaLocatio
ader
Schema Name: NEI
d:documentation>
tation>Current Version
http://www.epa.gov/exchange
n>
ntation>Description: The NEI
nat</xsd:documentation>
ntation>Application: Varies by
:documentation>
ntation>Developed By: Environme
ng="UTF-8" ?>
http://www.epa.gov/exchange
www.w3.org/2001/XMLSchema"
www.epa.gov/exchangenetwork"
="qualified" attributeFormDefault="unq
Location="EN_NEI_Common_v3_0.xsd
n>Schema Name: NEI XML 3.0
umentation>
n>Current Version
//www.epa.gov/exchangenetwork<
>Description: The NEI XML 3.0 Point
t:documentation>
Application: Varies by
tation>
```

Namespace Organization, Naming and Schema File Location

Version: 1.10

Revision Date: January 12, 2006

Prepared for: NTG

Prepared by:
Ross & Associates Environmental
Consulting, Ltd.



1	Introduction.....	3
2	EN Namespace Background	4
2.1	XML Namespaces	4
2.2	Namespace Naming: URL vs. URN.....	5
2.3	Decision to Recommend URL Formatted Namespaces for the EN	6
3	EN Namespace Recommendations	7
4	EN Namespace Technical Specification.....	8
4.1	Declaring EN Namespaces.....	8
4.1.1	Defining an EN Namespace	8
4.1.2	Namespace Construction.....	9
4.1.3	Namespace Sub-Categories	9
4.1.4	Namespace Prefix Guidelines.....	10
4.1.5	Namespace Versioning and Permanence	10
4.1.6	Reserved Shared Schema Components Repository Location	10
4.2	Instance File Namespace Declaration.....	11
4.2.1	Schema Location Tag in Instance Files	11
5	Implementing URL-based Namespaces	12
5.1	Recommendations for Schema Development.....	12
5.1.1	Single Root Schema per Namespace	12
5.1.2	Standardized Name for Default File	12
5.1.3	No Root Schema or Default File for Shared Schema Components.....	13
5.2	Recommendations for Repository File Structure.....	13
5.2.1	Default Delivery of Root Schema.....	13
5.2.2	Direct Registry Browsing	14
5.2.3	Extensibility.....	14
5.3	Recommendations for Shared Schema Components & Customized Flow Components	14
5.3.1	Shared Schema Components Namespace	14
5.3.2	Versioning of Shared Schema Components.....	15
5.3.3	Customized Flow Components.....	15
5.4	Recommendations for Versioning of Schema.....	16
5.5	Recommendations for Review of Flow Identification Parameter	16

Note: The key recommendations of this document are summarized in Section 3, with full technical specifications in Section 4. The preceding sections provide background and context for this guidance.

1 Introduction

As indicated in the *XML Design Rules and Conventions for the Environmental Information Exchange Network* (DRC)¹ rule SD4-13, all Exchange Network (EN) “data centric” schemas **must** declare a target namespace. However, namespace implementation remains a controversial topic in the XML community at large, and the current EN XML Registry confirms that several naming conventions are in use now (see Appendix 1).

These recommendations provide developers with guidance on selecting and organizing namespaces for their EN schema. These recommendations are intended to bring predictability to developers and users, and to resolve known points of confusion and implementation problems. They should be implemented as the “standard Network approach” unless superseded by documented, flow-specific requirements. These recommendations will stand until broader governmental and industry namespace best practices are developed, proven, and then adopted by the EN governance.

Recommendations provided here apply to current and future schema development. Additional guidance has been developed for XML instance file creation. Few schemas in the Registry are currently conformant to the recommendations outlined in this document. Schema developers should update the namespaces of existing schemas consistent with current recommendations at their next business-driven revision cycle.

Certain aspects of the present guidance are in conflict with existing EN documentation. In each of these cases, this document supersedes older recommendations. See Appendix 2 for details.

¹ The *XML Design Rules and Conventions for the Environmental Information Exchange Network* can be accessed through the EN website at http://www.exchangenetwork.net/dev_schema/. Note that the DRC will likely be revised to be made consistent with these namespace recommendations and additional developer feedback.

2 EN Namespace Background

2.1 XML Namespaces

An **XML namespace** is a collection of names, identified by a URI reference (i.e., either a URL or a URN)², which are used in instance files as element types and attribute names. The `targetNamespace` attribute is required for EN schemas. An example target namespace declaration:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://mydomain.org/schema/medical/1_0/">
```

Instance files can indicate which namespace a given tag name is being drawn from through the use of a two-part naming convention, where a prefix selects the namespace. In the example below, the element type `glasses` is drawn from the namespace associated with the namespace prefix `med`.

```
<med:glasses>...</med:glasses>
```

In order to use a namespace prefix, as in the example above, instance files must declare a namespace:

```
<x xmlns:med='http://mydomain.org/schema/medical'>
  <med:glasses>...</med:glasses>
</x>
```

The `xmlns` attribute allows the instance file author to bind a namespace prefix to a namespace (identified here by a URL). Namespace prefixes give unique pointers, and sometimes human readable hints, to the origin of the tags (e.g. indicating their affiliation with the EN and a given program). In the example given, the `med` prefix loosely conveys that the `glasses` name is drawn from a schema governing XML representations of medical information. More importantly, namespace prefixes allow the same name from multiple namespaces to be used in a single instance file. In the following example, two element types with the same name are used, but ambiguity is avoided through the use of namespace qualified names:

```
<dossier
  xmlns:med='http://mydomain.org/schema/medical'
  xmlns:obj='http://mydomain.org/schema/objects'>
  ...
  <med:glasses>wears graduated bifocal lenses</med:glasses>
  <obj:glasses>owns 6 fine crystal wine glasses</obj:glasses>
  ...
</dossier>
```

For more information, please refer to the W3C.³

² The acronyms URI, URL, and URN stand for, respectively, Uniform Resource Identifier, Uniform Resource Locator, and Uniform Resource Name. See http://en.wikipedia.org/wiki/Uniform_Resource_Identifier for definitions of these terms.

³ See "Namespaces in XML" at <http://www.w3.org/TR/REC-xml-names/#sec-intro>.

2.2 Namespace Naming: URL vs. URN

Two patterns have emerged in the developer community for namespace identification: Uniform Resource Name (URN) formatted namespaces and Uniform Resource Locator (URL) formatted namespaces. URNs were designed to be “persistent, location-independent, resource identifiers.”⁴ URNs are syntactically formatted starting with the top level domain and are separated by colons. URNs can be formally registered with the Internet Assigned Numbers Authority (IANA). Formal registration with IANA can permit a compliant application to resolve a URN to an internet location. URLs are familiar to most internet users. They are designed to convey a unique human-readable name as well as a machine-resolvable internet location for applications.

Debate over the use of URL versus URN for the “naming” of namespaces is ongoing in the XML developer community. Even among EN developers there is no agreement on a single preferred approach. URN formatted names (as in `urn:us:net:exchangenetwork:...`) were required in the original DRC but not widely implemented.⁵ URL formatted names (as in `http://www.exchangenetwork.net/...`) share a similar format to URN names, but may also support the location of resources on the Internet.

The W3C XML 1.1 standard is ambiguous regarding the use of URNs or URLs.⁶ While the W3C does recommend that namespaces be unique and persistent, compliance with the W3C XML 1.1 standard does not require namespaces to support direct schema retrieval.

Given that the W3C does not make a clear recommendation in this area, and the lack of consensus in usage on the EN, the Network Operations Board (NOB) elected to “not let the perfect be the enemy of the good” and to establish **one** EN approach to namespace construction, rather than to allow the current proliferation of approaches to continue.

As schemas and their tags continue to proliferate, the consistent use of namespaces on the EN will become increasingly more important. Given the immaturity of both the underlying conventions and our experience with them, these recommendations seek to balance several, sometimes competing, goals:

- Provide a means for uniquely identifying individual XML constructs used across the EN;
- Begin using Namespaces to manage XML constructs intended for Network-wide re-use.
- Manage the customizations/modifications of existing XML tag sets when they are deployed in other flows;
- Give developers and users simple guidance on namespace construction to avoid naming conflicts and confusion.

⁴ Moats, R. “URN Syntax” <http://www.ietf.org/rfc/rfc2141.txt>.

⁵ See http://www.exchangenetwork.net/dev_schema/xml_drc_section2.pdf page 29 and Appendix 1 for implementation information.

⁶ <http://www.w3.org/TR/xml-names11/#dt-NSName>.

2.3 Decision to Recommend URL Formatted Namespaces for the EN

The ability to resolve URL formatted names is seen as a major advantage by some because it can provide users and applications with a standardized way of naming and locating schemas. Out of deference to both past practice and a lack of consensus in larger standards/developer settings, the proposal to allow both URN **and** URL formatted namespaces was strongly considered but, in the end, rejected as needlessly complex. URLs are familiar to most Internet users and their presence in a namespace implies a human-readable name and a physical location. Interested readers can do a web search on “namespace URL URN” for more information.

Standardizing on URL formatted namespaces is recommended for the following reasons:

- URL formatted namespaces are simple and intuitive to internet users and developers
- URL formatted namespaces meet requirements set out by the W3C:
 - By their nature, URLs refer to a unique internet location
 - Within the scope of the EN, URLs pointing to `http://www.exchangenetwork.net/schema` will be persistent
- Namespaces will be unique to flows determined by the inherent structure of the repository
- URL namespace resolution is universally supported by all web applications:
 - Secondary registration is not required with IANA, reducing administrative overhead
 - On-the-spot resolution prevents naming errors – users get instantaneous feedback if they misspell a namespace by accessing the URL as a hyperlink
- The ability to resolve namespaces to a network location within the EN repository supports other EN priority initiatives, including:
 - Schema management and versioning
 - Improving the Schema Repository
- Standardized implementation of URL formatted namespaces supports future development initiatives based upon resolution of namespaces (e.g. RDDDL⁷)

⁷ See <http://www.rddl.org/>

3 EN Namespace Recommendations

The new standard for EN namespaces proposed in this document is summarized by the following three recommendations:

Recommendation 1: Namespaces on the EN will be URL formatted following the specifications outlined in section 4.1.2 “Namespace Construction.”

Recommendation 2: Version information is required in all EN namespaces. Accordingly, versioning of schema themselves must be strictly enforced.

Recommendation 3: When declaring EN namespaces in instance files, the use of the `xsi:schemaLocation` is optional. This tag can serve a subsidiary function for specifying schema against which instances can be validated. Further specifications are outlined in section 4.2.1.

DEPRECATED

4 EN Namespace Technical Specification

Schema developers should henceforth adopt the specific URL based namespace convention outlined in this document. This means that developers who have (as per the existing guidelines) used a URN formatted namespace will be required to update those namespaces at the next business-driven schema revision, resulting in iteration of the major version number. Existing schema now in use are “grandfathered” in. At the next major schema revision, any older versions that use URN formatted namespaces will be considered deprecated⁸.

The only exception to this guideline regarding gradual implementation of the namespace strategy is Shared Schema Components. Because of their widespread use, it is recommended the Shared Schema Components be updated to reflect this namespace decision as soon as possible (see Section 5.1.3 and 5.3).

4.1 Declaring EN Namespaces

The decision to use URL formatted namespaces gives users the ability to obtain useful information by resolving namespaces. Because EN namespaces will point to a commonly managed web server (at www.exchangenetwork.net/schema/), namespaces and the directory structure of the repository will be interlinked.

Based upon this recommendation, all EN schema must declare their target namespaces (with the `targetNamespace` tag) using URL formatted namespaces according to the guidance that follows in section 4.1.2. Likewise, when referencing EN schema, EN instance files must declare namespaces according to the specifications below. These specifications only apply to EN namespaces. Non-EN schema and namespaces are not affected by these requirements.

4.1.1 Defining an EN Namespace

The EN namespaces are defined as the set of schemas used on the Network which:

- are registered on the EN namespace registry, with an identified Namespace and Schema steward;
- have been constructed with a good-faith effort to conform to the applicable EN XML conventions and data standards (and documented in a Schema Conformance Report); and
- are managed such that all schemas within a category are constructed to avoid unintended naming conflicts when included in a root schema for that category.

EN namespaces are designated by using the following namespace root:

```
http://www.exchangenetwork.net/schema/
```

This namespace root is reserved for EN schemas. All EN namespaces should be a fully qualified (absolute) URL. Relative URLs are not permitted, as their use in namespaces violates W3C standards. Although relative URLs may resolve correctly, the document does not take on the

⁸ Recommendations on versioning of schema appear in section 5.3.

fully qualified URL as the namespace; instead the namespace remains the relative URL, which could lead to naming collisions.

4.1.2 Namespace Construction

Namespaces provide a mechanism to group related schema components together. Transactions on the EN are commonly grouped by the term “flow.” Each flow has a name, usually contains multiple schema files, and may have “sub-flows,” all documented in a Flow Configuration Document⁹. Here, instead of the word “flow,” the more generic terms “category” and “sub-category” are used.

Categories are typically named after the sponsoring organization name, subject area, or legacy system. Schemas within a single category are intended for widespread direct use by the relevant partners to accomplish a common set of business transactions. Category names must be unique across the EN, but **no** attempt is made to pre-define namespace groupings such as “air,” “water,” or “waste,” since there is no consensus on a single appropriate hierarchy.

For the purposes of managing the EN namespaces, the key function of category and sub-category names is to provide users with a unique name associated with a set of schema files.

EN namespaces will be constructed as follows:

`http://www.exchangenetwork.net/schema/category[/sub-category]/version`

category	Category name is unique across the EN and identifies the transaction or exchange.	Required
sub-category	A further sub-division meaningful to those participating in the transaction or exchange (these only need to be unique within the category).	Optional
version	Schema version information (includes major and minor version numbers, separated by an underscore). See section 5.4 for discussion of versioning.	Required

Examples:

`http://www.exchangenetwork.net/schema/AQS/3_2`

`http://www.exchangenetwork.net/schema/RCRA/Handler/1_1`

4.1.3 Namespace Sub-Categories

Creation of sub-categories is appropriate, and strongly recommended, whenever parts of a flow constitute independent transactions governed by separate schema. As implied by the

⁹ See <http://www.exchangenetwork.net> for example Flow Configuration Documents.

requirements above, sub-categories (within a given flow or category) are versioned separately and represent independent namespaces¹⁰.

If flow-specific schema components are shared across sub-categories, it is recommended these components be placed in their own versioned sub-category (and hence their own sub-directory, and namespace, within the flow). For example, if the RCRA flow has several sub-categories that share a schema file or files specific to RCRA, these should be placed in their own sub-category.

4.1.4 Namespace Prefix Guidelines

Namespace prefixes can be any short abbreviation for the category. Although arbitrary, they should remain consistent within the transaction or exchange¹¹. The following guideline applies:

- Prefix xsl, xsd, xs xsi, and other standard schema related prefixes are also reserved for their respective standard uses.

4.1.5 Namespace Versioning and Permanence¹²

Once published, the tag sets (and namespace) defined by a schema are permanent. Changes to published schemas **MUST** be indicated by version changes per the guidelines in section 5.4. Since version information is required on namespaces, the iteration of version number on a schema necessarily defines a new namespace. Developers working within a finite and known community may elect to be more lax about iterating the schema version and target namespace as changes are made to “draft” schema.

4.1.6 Reserved Shared Schema Components Repository Location

EN Shared Schema Components are standard element and type definitions intended for broad re-use in EN Schema. As discussed in Section 5.3, Shared Schema Components should not declare a target namespace in order to better facilitate their broad use. However, the following location in the Repository is reserved for the Shared Schema Components:

`http://www.exchangenetwork.net/schema/sc/`

Details on the usage of Shared Schema Components can be found in the *Exchange Network Shared Schema Components: Usage Guide*.¹³ Note that version 1.0 and 1.1 of these components

¹⁰ A new namespace is created with any unique string, so the namespace of a sub-category is truly its own namespace. The notion of sub-categories is useful, nonetheless, to indicate a relationship between namespaces that share a common category root. However, XML Namespace aware tools will not enforce name uniqueness across sub-categories. Doing so requires other processes.

¹¹ Software tools vary in their “awareness” and consistent interpretation of namespace prefixes. In some cases, when customizing a flow schema, developers may choose to **not** change the prefix of the imported elements, but only the namespace to which it refers. This “trick” may provide compatibility with “prefix aware” stylesheets and other applications, and require only a single change to the namespace reference (besides those required in the customization itself) to software components developed for the original flow schema.

¹² These versioning rules are adopted from those developed by the SAML specification:

<http://xml.coverpages.org/saml.html>

¹³ Available on the EN website at <http://exchangenetwork.net/registry/SharedSchemaComponents-UsageGuide.pdf>

were developed prior to the adoption of this recommendation. It is recommended that the Shared Schema Components be revised as soon as possible to adopt the changes proposed in Section 5.3.

4.2 Instance File Namespace Declaration

Instance files on the Exchange Network must use the `xmlns` tag to specify the namespace(s) from which tag names have been drawn. Generally, instance file declarations of EN namespaces should be identical in form to the target namespace declarations found in EN schema. The only exception is referencing of Shared Schema Components, which do not declare a namespace and can be included directly. See the discussion in Section 5.1.3 and 5.3.

4.2.1 Schema Location Tag in Instance Files¹⁴

The “[XML Schema Part 1: Structures Second Edition](#)” specifies the attribute `schemaLocation` for defining schema location. This tag is an optional element in the EN strategy for declaring namespaces in instance files. The `schemaLocation` tag allows developers to assert a specific schema file against which they believe their instance file can validate. Validation against files specified in `schemaLocation` is not guaranteed, and, for schemas with an EN namespace, use of the `schemaLocation` tag is generally redundant. Some organizations have chosen to disable `schemaLocation` processing due to the potential for linking of malicious files.

¹⁴ Note that these comments do not apply to use of the `schemaLocation` tag in schema files, where, for example, `schemaLocation` is required in an `Include` element.

5 Implementing URL-based Namespaces

This section considers implementation issues related to the EN Namespace specification above. These new specifications have implications for schema development, structure of the schema repository, use of the Shared Schema Components, versioning, and flow identification.

5.1 Recommendations for Schema Development

Meeting the recommendations for URL formatted EN network namespaces implies the following requirements for schema developers:

- Default delivery of the root schema file for each namespace
- Maintenance of a one-to-one correspondence between a namespace and a schema file, the root schema

5.1.1 Single Root Schema per Namespace

A one-to-one mapping between a namespace and a single schema file is necessary to support the unambiguous resolution of URL formatted namespaces. Therefore, even for a namespace that contains many schemas within it, a root schema must be created that includes all definitions within that namespace.

Following standard EN naming convention, root schema shall be named as follows:

```
category[_sub-category]_version.xsd
```

Providing developers create root schema and all relevant schema are included in this file, then recipients should be able to validate any instance file using that EN namespace. Note that a root schema does not itself need to be a message schema; however, it is required to provide a single entry point for all definitions in that namespace. Rules regarding default namespaces and namespace coercion from the EN Design Rules and Conventions (rule 2.4-10) are not superseded by this recommendation.

5.1.2 Standardized Name for Default File

Another integral part of this namespace strategy is the default delivery of the root schema to applications attempting to resolve an EN namespace. As specified in Section 4 above, namespaces locate a directory, not a specific schema file. Therefore, in order to support default delivery of the root schema, a default file with standardized filename must be provided for each namespace. This file will be named:

```
index.xsd
```

The default file simply includes the root schema for that namespace. This file allows the recipient to load all definitions for a namespace without figuring out the filename of the root schema. Additionally, it allows human viewers of the repository to easily locate the root schema by accessing the address of its parent directory.

5.1.3 No Root Schema or Default File for Shared Schema Components

All versioned folders in the EN repository that correspond to a valid EN namespace will contain a root schema and an `index.xsd`. Note that Shared Schema Components do not declare a target namespace, and therefore the repository folder containing them will have no root schema or default file¹⁵.

For example, the folders corresponding to the following URLs would contain a root schema and an `index.xsd`:

```
http://www.exchangenetwork.net/schema/AQS/3_2
http://www.exchangenetwork.net/schema/RCRA/Handler/1_1
```

These, however, would not¹⁶:

```
http://www.exchangenetwork.net/schema/AQS/
http://www.exchangenetwork.net/schema/RCRA/
http://www.exchangenetwork.net/schema/RCRA/Handler/
http://www.exchangenetwork.net/schema/SC/2_0/
```

5.2 Recommendations for Repository File Structure

The recommendations for URL formatted EN network namespaces have direct bearing on the structure of the EN repository. Therefore, recommendations have been developed to support the following goals:

- Default system delivery of the default file that includes the root schema for that category (or sub-category) for applications such as validating parsers that will need to resolve namespaces.
- The ability for internet users to browse the repository with a standard web browser.
- Providing extensibility to accommodate future changes in directory structure and growth of categories and sub-categories.

5.2.1 Default Delivery of Root Schema

To support delivery of the root schema, the default document served by the web server for the virtual directory `http://www.exchangenetwork.net/schema/` will be `index.xsd`, the default file specified in 5.1.2. The web server delivers the default file in response to all http requests that are not fully specified (i.e., do not end with a file name). This default document will be specified for the entire virtual directory, though only certain folders will contain an `index.xsd` file, as shown in the examples in section 5.1.3.

¹⁵ Due to the large number of Shared Schema Components, it is inefficient to load all of these tag name definitions at once. Developers must individually include the specific components they need.

¹⁶ The first three folders should not contain any schema files at all since they are not well-formed namespaces according to this recommendation. The fourth folder does not contain an `index.xsd` because there is no root schema for the Shared Schema Components.

5.2.2 Direct Registry Browsing

The recommendation in preceding sections regarding the implementation of root schema and default file supports the automatic delivery of an appropriate schema file to xml-aware applications. However, this conflicts with the desire to allow direct (human) browsing of the repository using a standard web browser. Both needs can be satisfied by simply creating an additional virtual directory, such as:

```
http://www.exchangenetwork.net/repository/schema/
```

The directory maps to the same physical location, but does not have a default document specified (and allows directory browsing).

The proposed repository structure will support the ability for browsing the repository using standard browsers and alleviate the potential for web browsers being served the `index.xsd` file when attempting to view the contents of a directory. Maintaining a virtual directory designed for human access also provides flexibility as underlying changes in the repository structure can be accommodated (transparently to flow developers and users) through additional or modified virtual directory mappings.

5.2.3 Extensibility

The namespace recommendations contained within this document are compatible with future growth and change in the EN. Embracing URL formatted namespaces does not unnecessarily constrain the directory structure of the network repository. Examples in this document suggest the most logical use of category-based namespaces informed by existing EN practices. However, the decision to implement URL formatted namespaces retains a great deal of flexibility. The W3C requirement that namespaces be unique and persistent (see section 2.2) is integrally enforced through the use of URL-specified namespaces. Virtual directory mappings can ensure backwards-compatibility and persistence in the event of future changes in the repository directory structure.

5.3 Recommendations for Shared Schema Components and Customized Flow Components

5.3.1 Shared Schema Components Namespace

The recommendations in this section supersede the Shared Schema Components Usage and Technical Guides regarding repository location and namespace for SSCs. These recommendations are designed to conform to other guidelines in this document while not limiting how developers incorporate SSCs into schema files.

- Managed SSCs—those listed in the SSC usage guide—will not declare a target namespace. This recommendation facilitates schema developers using `include`, *rather than import*, when using SSCs in their schema (include has technical advantages over

import). These SSCs will be located in the directory:
<http://www.exchangenetwork.net/schema/sc/>

- Other schemas designed for reuse and sharing, typically those developed for particular flows, may declare target namespaces to prevent naming conflicts with other schema.
- SSCs may be placed in sub-categories (with a corresponding directory structure) under the root sc directory: <http://www.exchangenetwork.net/schema/sc/>
- SSCs will not have a root schema, except in the case where they are given a namespace to avoid naming conflicts. When SSCs do declare a target namespace, then there must be a root schema to facilitate importation into other schema (i.e., the one namespace–one root schema rule must be abided: either shared schemas declare no namespace and have no root schema, or they declare a namespace and have a root schema).

5.3.2 Versioning of Shared Schema Components

It is not expected that the Shared Schema Components will be updated frequently. However, when an individual SSC does need to be updated, the other SSCs (and schema files that include the other SSCs) need not be affected. Because SSCs are included by schema files individually, a new version of the AgencyType SSC, for example, should not require that the version number of the GeometricType SSC (or any other SSC) be incremented. In order to allow for the versioning of individual SSCs, while keeping all SSCs in the same directory making it easy for developers to find the latest version of each SSC, it is recommended that the SSCs simply be placed in the folder corresponding to the location reserved for the SSCs in Section 4.1.6, without an additional layer of folders for versioning.

Therefore, the three files below (with 2.1 being the latest version of AgencyType and 2.0 being the latest version of GeometricType) would all reside in the same directory:

http://www.exchangenetwork.net/schema/sc/AgencyType_v2.0.xsd
http://www.exchangenetwork.net/schema/sc/AgencyType_v2.1.xsd
http://www.exchangenetwork.net/schema/sc/GeometricType_v2.0.xsd

5.3.3 Customized Flow Components

The namespace recommendations made in this paper embrace the design principles that were developed (and are detailed) in the *Shared Schema Components: Usage Guide*. They describe how namespaces and element/type (constructs) names should be used together when modifying an xml construct. This is summarized in the table below:

Re-Use Scenario	Namespace	Element/Type Name
Compatible Customization ¹⁷	Target flow name	Original
Re-use but with Significant Changes	Target flow name	New derived name
Major modifications	Target flow name	New original name

¹⁷ See definition in the *Shared Schema Components: Usage Guide* (available at <http://exchangenetwork.net/registry/SharedSchemaComponents-UsageGuide.pdf>).

5.4 Recommendations for Versioning of Schema

Comprehensive versioning guidance has not yet been developed for the Exchange Network. The EN XML Design Rules and Conventions (DRC) document provides rules¹⁸ governing schema versioning, which are not superseded, by this paper. An NTG Task Force has been convened to develop guidelines to help the EN implement versioning. The following recommendations are in accord with basic versioning principles and the rules existing in the DRC:

- Major versions should represent fundamental changes to the information being versioned that do not imply a possibility of compatibility in syntax, semantics, or implementation. Higher major versions may be a superset or a subset of functionality present in lower major versions.
- Minor versions imply backward compatibility and should represent less significant changes to the information being versioned. Instance files should be able to validate against subsequent minor versions.

A primary motivation for incorporating versioning in EN namespaces is providing a complimentary mechanism for managing the increase of final schemas in use now. As flow groups create new elements and “add” these to their namespace, they can increment the minor versions. This allows users of previous versions to continue operations, but clearly signals, and makes available, the new elements to those who are interested. The issue of *what* versions of schema are supported by *who* is a separate flow-specific policy issue.

5.5 Recommendations for Review of Flow Identification Parameter

To allow rapid routing of EN data, a Data Flow parameter is provided in the XML header. Though currently used only by CDX, this parameter allows nodes to quickly determine the type of data contained in an EN message payload. This provides significant performance gains in cases that would otherwise require parsing a large payload only for the sake of determining which flow the message belongs to.

New EN namespaces as specified here provide a solid basis for standardizing the content of this Data Flow parameter. One possibility is to adopt a flow identification scheme similar to the specification of root schema filenames:

```
category[_sub-category]_version
```

This issue will be addressed in due course by the NTG, perhaps as part of a general review of the XML header specification, resulting in an update of the Exchange Network Design Rules and Conventions.

¹⁸ See http://www.exchangenetwork.net/dev_schema/xml_drc_section2.pdf_section_2.5_pages_11-12. SD5-20 and SD 5-21 which require a major and minor version number included in data-centric schemas using the W3C Schema version attribute.

Appendix 1. Current namespace definitions on EN

Schema	Version	Namespace
AQS	1.0	urn:us:net:exchangenetwork:AirQuality:AirQualitySubmission:1:0
ABATE	0.5	none
Beach Monitoring	1.2	none
Beach Notification	1.0	none
Data Services EN	1.0	urn:us:net:exchangenetwork
e-DMR	1.2	none
e-DWR	2.0	urn:us:net:exchangenetwork
Asbestos	2.0	http://www.epa.gov/xml
Exchange Network Document Header	0.9	http://www.exchangenetwork.net/schema/v1.0/ExchangeNetworkDocument.xsd
FACID	2.3	http://www.epa.gov/xml
ISIS	1.0	http://www.exchangenetwork.net/schema/draft/isis
IDEF	1.3	none
NEI	3.0	http://www.epa.gov/exchangenetwork
OWWQX	1.0	urn:us:net:exchangenetwork:owwqx:1:0
PNWWQX	1.3	urn:us:net:exchangenetwork
RCRA	1.0	urn:us:net:exchangenetwork:RCRA:RCRA_C:1:0
SDWIS	2.0	none
SSC / CRM	1.0	urn:us:net:exchangenetwork:sc:1:0
TRI	1.1	urn:us:net:exchangenetwork:TRI:1:1

Retrieved From: [http://iaspub.epa.gov/emg/xmlsearch\\$.startup](http://iaspub.epa.gov/emg/xmlsearch$.startup) - 11/16/05

DEPRECATED

Appendix 2. EN documents which will need to be updated to comply with this Namespace recommendation:

Data Exchange Design Guidance and Best Practices Document v 1.0

http://www.exchangenetwork.net/dev_schema/exchange_design_guidance_v1.0.pdf

No Changes Required

XML Design Rules and Conventions v 1.0

http://www.exchangenetwork.net/dev_schema/xml_drc_section2.pdf

2.4-1 - 2.4-2 (p 29-30)

Change requirement from URN to URL-formatted namespace.

URL-formatted namespaces on EN must resolve to a valid web address

2.4-5 - 2.4-7, 2.4-9 (p 33-35,37)

Example should be changed to URL-formatted namespace

2.4-13 (p 42)

SD4-28/SD4-29 Must be removed. New rules should be added to comply with namespace guidelines contained in this report.

2.5-11 (p 59)

EN Schema Versioning Section should have guidance and rules added to include versioning guidance in namespace document

A-9 (p 105)

SD4-23/SD4-24/SD4-28/SD4-29 should be examined and, if appropriate, modified to comply with new namespace guidelines

Shared Schema Components Usage Guide v. Last Call Working Draft

<http://www.exchangenetwork.net/registry/SharedSchemaComponents-UsageGuide.pdf>

4.2.1.1 (p 17)

Change fixed namespace for Shared Schema Components

4.3.1, 4.3.2, 4.3.3, 4.3.4, 4.3.5, 4.3.6, 4.3.7, 4.3.8, 4.3.9 (p 19,21, 24-25, 27, 29-30, 31, 35-36, 38-39, 41-42)

Examples should be changed to reflect section 5.1.3 and 5.3

5.1 - 5.10 (44-54)

Examples schemas should be changed to URL-formatted namespace

Shared Schema Components Technical Reference v. Last Call Working Draft

<http://www.exchangenetwork.net/registry/SchemaComponents-TechnicalReference.pdf>

All pages

Examples should be changed to reflect section 5.1.3 and 5.3

Core Reference Model v 2.0

Pending NOB Approval