

November 5, 2010

DOMAIN UPDATE SPECIFICATION

VERSION 1.0

URI: <http://reference.niem.gov/niem/specification/domain-update/1.0/>

Change History

No.	Date	Reference: All, Page, Table, Figure, Para	A = Add M = Mod D = Del	Revised By	Change Description
1.0	2010.11.05				Initial release

Contents

Contents	ii
1 Introduction.....	1
1.1 Scope.....	1
1.2 Audience	1
1.3 Terminology.....	1
1.3.1 RFC 2119 Terminology	2
1.3.2 NIEM Terminology.....	2
1.4 Normative References.....	2
2 Domain Update as a NIEM Model Package Description	2
2.1 IEM Descriptions.....	2
2.2 Differences from the MPD Baseline Specification.....	3
3 Types and Usage.....	4
3.1.1 Harmonization of Domain Updates.....	4
3.1.2 Coordinated Domain Update.....	5
3.1.3 Incremental and Replacement Schema Domain Updates	6
3.1.4 Other mechanisms for domain changes	6
4 Domain Update Package.....	6
4.1 Change Schemas	6
4.1.1 Scope of Domain Changes	7
4.1.2 Change Schema Content	8
4.1.3 URI Syntax for Change Schema Target Namespace	9
4.1.4 Other XML Schema Artifacts	10
4.2 Catalog.....	10
4.2.1 URI Syntax for MPD.....	10
4.2.2 Artifact Nature and Purpose	11
4.2.3 Lineage	12
4.3 Change Log.....	13
4.4 Quality Assurance Reports	14
4.5 Sample Instances.....	15
Appendix A: Publication Area.....	A-1
A.1 Publication Area Name	A-1
A.2 Publication Area Content.....	A-1
A.2.1 Change Schemas	A-1
A.2.2 Change Logs.....	A-2
These rules are also applicable for coordinated domain updates.	A-3
A.2.3 Domain Update Packages.....	A-3
A.2.4 Coordinated Domain Updates	A-3
A.2.5 Prerelease Domain Updates	A-4
A.3 Publication Area Service Interface	A-4
A.4 Multiple Publication Areas	A-4
A.5 Future Versions.....	A-5
A.5.1 Application Program Interface (API)	A-5

Appendix B: Sample Schemas for Incremental Update and Replacement Update	B-1
B.1 Incremental Schema Domain Update	B-1
B.2 Replacement Schema Domain Update	B-2
Appendix C: Glossary.....	C-1
Appendix D: Acronyms and Abbreviations.....	D-1
Appendix E: References	E-1

Figures

Figure A-1 - Interaction between Publication Areas	A-5
Figure B-1 - Incremental Domain Update	B-1
Figure B-2 - Replacement Domain Update	B-3

1 Introduction

The National Information Exchange Model (NIEM) High Level Version Architecture (HLVA) [NIEM-HLVA] defines a domain update as both a process and a product. As a process, a domain update identifies issues and requirements for domain content based on analysis of NIEM releases, Information Exchange Package Documentations (IEPDs), and Enterprise Information Exchange Models (EIEMs). As a product, a domain update allows a domain to resolve those issues and satisfy those requirements. This document elaborates on the domain update process and provides a normative specification for the domain update product.

1.1 Scope

This specification builds on the NIEM Model Package Description (MPD) Specification [NIEM-MPD], and provides both normative rules and non-normative guidance for the packaging, content, and publication of domain updates. This specification applies to any MPD that indicates it conforms to this specification. The provisions of this document become effective after the conclusion of a successful test period and subsequent approval by the NIEM Program Management Office (PMO). If a domain update indicates that it is conformant to this specification, then it must also be a conformant MPD as defined by [NIEM-MPD].

This revision of the specification does not address the domain update process as it applies to code lists; however, upon approval of the NIEM Code List Specification [NIEM-CODE-LIST] by the NIEM PMO, the NIEM Technical Architecture Committee (NTAC) will release a revision of this specification that does address that process. This revision also does not address the domain update process as it applies to domains that reference external schema standards. A forthcoming revision of this specification will address this process.

1.2 Audience

This specification is meant for use by those who produce and consume domain updates. They include domain stewards, who produce domain updates to change their domain content; the release manager, who consumes domain updates to build a release; IEPD and EIEM developers, who consume domain updates to build IEPDs and EIEMs; and developers, whose tools assist the users.

This specification assumes that the reader understands NIEM concepts and rules defined in the NIEM Naming and Design Rules (NDR) [NIEM-NDR], the NIEM High Level Version Architecture [NIEM-HLVA], and the NIEM Model Package Description Specification [NIEM-MPD].

1.3 Terminology

This document uses standard terminology to explain the rules that define a domain update.

1.3.1 RFC 2119 Terminology

Within normative content (rules and definitions), the key words **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **MAY**, and **OPTIONAL** in this document are to be interpreted as described in **[RFC2119]**.

1.3.2 NIEM Terminology

Terms and definitions related to NIEM may be found in the Appendix C: Glossary.

1.4 Normative References

The following specifications constrain the behaviors used in this document.

[RFC2119] defines how specific words in this document will be interpreted.

[RFC5234] defines the Augmented Backus-Naur Form, which is used to define the URI Syntax for the `targetNamespace` of a domain and for the domain update package, including the following core ABNF syntax rules defined by that specification: DIGIT (decimal digits).

[NIEM-MPD] provides the basis for this document and the framework upon which this specification is built.

[NIEM-HLVA] describes the overall architecture of NIEM. This specification implements specific parts of that architecture.

[NIEM-NDR] specifies rules for XML schema documents when they are used in NIEM.

Of particular interest in the **[NIEM-NDR]** is the concept of schema-namespace correspondence. To simplify automatic schema processing and reduce the potential for confusion and error, **[NIEM-NDR]** Principle 8 states that “each NIEM-conforming namespace **SHOULD** be defined by exactly one reference schema.” To support this principle, **[NIEM-NDR]** Rule 6-5 forbids the use of `xsd:include`, and **[NIEM-NDR]** Rule 6-35 requires the use of the `xsd:targetNamespace` attribute in NIEM-conforming schemas.

The foregoing **[NIEM-NDR]** rules and principle imply that each NIEM namespace is defined by a single reference schema and each NIEM-conforming schema declares a target namespace.

2 Domain Update as a NIEM Model Package Description

This section is non-normative.

2.1 IEM Descriptions

Per **[NIEM-MPD]** a domain update is one of several classes of NIEM Information Exchange Models (IEM), each of which has a different purpose. An MPD is allowed to contain one and only one IEM. The type of an MPD is determined by the class of IEM it contains. For convenience, the IEM classes are summarized here:

- NIEM release – A full set of schema artifacts that coherently define all content within a single version of NIEM.
- Domain update – One or more schemas that constitute changes to a NIEM domain.
- Information Exchange Package Documentation (IEPD) – A set of schemas that define a recurring NIEM data exchange.
- Enterprise Information Exchange Model (EIEM) – A NIEM-conforming schema set that subsets and extends NIEM. It may also include formal business rules. An EIEM is used to build consistent IEPDs for an enterprise.

Per [NIEM-HLVA], using NIEM numbered releases and domain updates, IEPD and EIEM developers identify new requirements. A NIEM domain can use these new requirements to build content improvements and extensions between NIEM releases, and to propose changes to future NIEM releases. The product of this process is a *domain update*.

The remainder of this specification will generally refer to a domain update as a Model Package Description (MPD) that contains a domain update IEM as well as required documentation and metadata.

2.2 Differences from the MPD Baseline Specification

The [NIEM-MPD] defines three categories of artifacts that a MPD may include: (1) XML schema artifacts include reference schemas, subset schemas, extension schemas, exchange schemas, and constraint schemas; (2) Mandatory documentation artifacts include a catalog and change log; (3) Other artifacts (usually optional) can include sample instances, additional documentation, and tool support files.

This Domain Update specification requires that a domain update include the following mandatory artifacts:

1. Change schema – a reference schema or reference schema set.
2. Catalog – records all artifacts and associated semantics and metadata.
3. Change log – records all changes (deletions can only be recorded in the change log)
4. Quality assurance report – records results of automated quality checks on a domain update.

Domain update change schemas contain the modified schema content for the domain. This specification requires change schemas to have URIs that follow a specific pattern. This allows the publication of the schemas to the correct locations to make them available to the public.

Rules that specify metadata in the catalog are defined later in this document. A domain update must specify the type of domain update, the NIEM release being updated, and the URI for the domain update package.

The [NIEM-MPD] also defines a change log to record all changes. A change log is necessary to assist governing bodies and the release manager during the harmonization and integration of updates into the next NIEM release. Instead of documenting changes from a previous version of a MPD, a domain update change log will document changes to domain content from a NIEM release and/or a previous domain update. The change log also assists IEPD and EIEM

developers in understanding specific changes and their potential impacts on IEPDs, EIEMs, and other domains.

A domain update is also required to include quality assurance reports to show passing of fundamental tests. A quality assurance report verifies that at the time it was published, the domain update conformed to all the applicable rules.

These additional requirements reflect the different purpose of a domain update relative to other IEMs. The purpose of a domain update is to update the content of a domain in the NIEM model. Therefore, subset, extension, exchange, and constraint schemas are irrelevant and not contained in domain updates.

3 Types and Usage

A domain update is an MPD package of artifacts that contains new and/or changed domain-specific content developed between NIEM release cycles. The domain update may be published to a publication area, where it is immediately available for use in IEPDs and EIEMs. A domain update contains schema and documentation artifacts including the change log. The *change schema* (or change schema set) is a [NIEM-NDR] reference schema conformance target and follows that format. *The change log* denotes all the changes to the domain made in this domain update and uses the format specified by [NIEM-MPD]. A domain update is also required to have a catalog and a quality assurance report, and may contain other optional documentation and sample instances.

Per the [NIEM-HLVA], the domain update process occurs in the span of time between a given major, minor, or micro release and the next. During this period a domain may publish a number of domain updates. Each of these updates represents another change in that domain namespace. These updates may be either *incremental*, where the domain update builds off a previous domain update, or full *replacement* where the content in the domain replaces the original domain content. The original domain namespace together with these updates represents a collection of NIEM data components that may be used in IEPDs and EIEMs. The *base NIEM release* is the release upon which the domain update is based, and the *previous domain update* is another domain update upon which a domain update is based. These define a domain update's relationships with other MPDs. For an incremental domain update, the previous domain update must be defined, while this is optional for a replacement domain update. The base NIEM release must be set for both types of domain updates, as it is an indicator to tools which NIEM version the domain is working within.

Certain domain updates may be used to create a micro release. This mechanism will be specified in a future version of this specification.

3.1.1 Harmonization of Domain Updates

A NIEM domain steward should use the domain update process to publish content changes arising from new requirements and/or resolutions of major bugs or modeling issues if IEPDs and EIEMs require them before the next release will be available.

Domain updates will accumulate in the publication area between NIEM releases. All domain updates published since the last major or minor release will be considered for integration into the

next release. For major releases these updates will be harmonized and integrated with the latest Core and corresponding primary domain namespaces. For minor releases these updates will be harmonized and integrated with only the corresponding domain namespaces.

Components from domain updates will be harmonized and integrated with existing NIEM content if it is a candidate for the next major or minor release. These changes will undergo review and harmonization by NIEM governance bodies. The NIEM Business Architecture Committee (NBAC) and the NIEM Technical Architecture Committee (NTAC) review these changes together with issues logged into the NIEM Configuration Control Tool (NCCT). The NBAC and NTAC review all issues, adjudicate, and harmonize the associated content. Subsequently, after socializing and reviewing the changes with the domain steward, the NIEM release manager integrates the content into the next NIEM major or minor release. In some cases, such as when update conflicts exist or additional changes are required, it may be necessary for the domain stewards to harmonize their own update components and to submit and updated change submission for release integration.

Note that a data component defined in a domain change schema is subject to harmonization with existing namespaces and may or may not remain in its original form. For a given domain, each change schema and its corresponding current primary domain schema will be harmonized and integrated into a single domain namespace. In major releases, some data components may be promoted into, harmonized with, and integrated into NIEM Core. All resulting changes to NIEM are vetted through NIEM governance bodies.

The domain update process also serves as an alternative mechanism for requesting changes to future NIEM releases. Normally domain stewards prepare and submit content *change requests* during a major or minor release cycle. Unlike a change request, a domain update does not require adjudication and harmonization before it is available for use. A domain steward must merely ensure that a domain update conforms to applicable rules in the NIEM Naming and Design Rules (NDR) [NIEM-NDR], the [NIEM-MPD], and this Domain Update Specification. This allows a domain steward to respond quickly to community needs.

A domain steward should use the domain update process with discretion and only for changes that are essential to IEPD or EIEM development. The reason for careful consideration is that domain updates may define different versions of semantically similar or identical components, creating added complexity for IEPD and EIEM developers.

This specification leaves the decision on the need for a domain update to the domain steward. The NBAC is available to offer more specific guidance to domain stewards if requested based on actual scenarios.

3.1.2 Coordinated Domain Update

It is also possible for multiple domains to work together and publish an interdependent domain update to reduce the potential effects that domain updates can have on dependent domains. This is referred to as a *coordinated domain update*. Each participating domain has a separate schema artifact, but all schemas will be published together in a single domain update package. Though similar to the required formal coordination across all domains during major and minor release cycles, this coordination is optional, less formal, and usually smaller in scope. Domain update coordination may occur among any two or more domains. However, the more cross-domain

namespace referencing that exists, the more that coordination of domain updates will ease the burden on all IEPD and EIEM developers who must understand and employ them.

Before a coordinated domain update may be published, each participating domain steward must authorize the update; in this way, no coordinated domain update may be published without the knowledge and approval of the affected domains. A future version of this specification will define the mechanism by which a domain steward authorizes a coordinated domain update.

3.1.3 Incremental and Replacement Schema Domain Updates

This specification allows a domain steward to publish either an incremental or replacement schema update. A domain update based on replacement schemas is a full schema set for the domain, and redefines those components that have not changed. An incremental schema domain update contains only the components that have been added or changed since the previous domain update.

Appendix A: provides a representative example of an incremental schema update and a replacement schema update.

The domain steward should consider whether an incremental or replacement schema update is easier to use by IEPD/EIEM developers. The domain steward can reduce the number of components that are unnecessarily duplicated by using the incremental schema update. A replacement schema update may be more appropriate when a significant portion of the domain has been modified.

The specification does not mandate if or when a domain steward should deploy an incremental or a replacement update. Examples of both types are shown in the appendices. However, a domain steward should carefully consider the potential complexities and impacts to all IEPD/EIEM developers who will use domain updates.

3.1.4 Other mechanisms for domain changes

A domain may also update its content with a change request [NIEM-CRT] or UML model [NIEM-UML]. A change request represents content in spreadsheet format; a UML model represents content in UML. This specification addresses only the domain update method.

4 Domain Update Package

This section is normative.

A domain update package is a specialization of an information exchange model (IEM).

Rule 4-1: A domain update package MUST be a conforming Domain Update as defined by [NIEM-MPD] and this specification.

4.1 Change Schemas

XML change schemas in a domain update are based on the MPD XML schema artifacts from Section 3 of the [NIEM-MPD].

4.1.1 Scope of Domain Changes

Within a NIEM release, a domain produces schemas for `targetNamespaces` allocated to it by the NIEM PMO. These namespaces have base URIs of the form:

<code>http://niem.gov/</code>	NIEM release versions 1.0, 2.0, 2.1
<code>http://release.niem.gov/</code>	NIEM release versions after 2.1

A domain steward only publishes changes to content in his or her domain. This prevents a domain from interfering with and/or forcing content changes to another.

Rule 4-2: A domain update package MUST NOT contain schemas from namespaces that are not allocated to it by the NIEM PMO. However, a domain update package MAY reference and extend content under the control of another domain via component reuse, extension or augmentation, as permitted by the [NIEM-NDR].

The rule above does not preclude the coordination of a set of updates by multiple domains. Nevertheless, each domain still prepares its own change schema.

The base NIEM release is identified with a `Relationship` element in the catalog artifact as defined by [NIEM-MPD]. The following is an example relationship entry in the catalog of a domain update that applies to NIEM 2.1:

```
<Relationship relationshipCode="updates"
  resourceURI="http://release.niem.gov/niem/2.1/"
  descriptionText="CBRN domain update to NIEM 2.1" />
```

Rule 4-3: A domain update package MUST declare one and only one base NIEM release.

Rule 4-4: A domain update package MUST not import namespaces from a NIEM release other than its base NIEM release.

Rule 4-5: A domain update package MUST NOT reference other published domain updates built from a different base NIEM release.

These rules also permit a change schema to reference previously published domain update change schemas, as long as they all apply to the same NIEM release. The base NIEM release for a domain update is transitive; domain updates built from each other always use, and therefore, apply to the same base NIEM release.

4.1.2 Change Schema Content

A domain update change schema represents an addition to or modification of a reference schema. As such, like the reference schema, a domain update change schema must conform to [NIEM-NDR] as a reference schema conformance target.

Rule 4-6: A domain update change schema MUST conform to the [NIEM-NDR] as a reference schema conformance target.

The content of the domain update change schema depends on whether the domain update is an incremental or replacement domain update.

In the case of an incremental domain update, the content of the domain update change schema need only include those components that the domain steward wishes to add or modify; the change schema may include those components that the domain steward wishes to remain unchanged, as the inclusion of those components may improve the readability of the change schema.

Rule 4-7: In an incremental domain update package, the domain update change schema MUST include those components added to or modified in the domain and MAY include any component unchanged from the domain.

While the domain steward may represent the addition and modification of components in an incremental domain update change schema, the domain steward cannot represent deletions. To do so, the domain steward must use the change log to indicate the deletion of a component from the domain.

Rule 4-8: In an incremental domain update package, the change log MUST indicate those components deleted from the domain.

In the case of a replacement domain update, the content of the domain update change schema must include every component that the domain steward wishes to include in the domain; any component absent from the change schema will be absent from the domain.

Rule 4-9: In a replacement domain update package, the domain update change schema MUST include every component in the domain.

Rule 4-10: In a replacement domain update package, the change log MAY indicate those components deleted from the domain.

With the exception of **Rule 4-3**, this specification does not restrict the scope of a change schema content for a given domain update. However, a domain steward should be judicious with respect to the modification of components, as such modification may result in IEPDs and EIEMs that include different versions of the same or similar components.

In the case of a coordinated domain update, each domain produces its own change schema, and the domain update includes a change schema from each domain. This ensures that quality

checks and conformance checked may be performed on every change schema, cross-domain imports may be synchronized across every change schema, and the change schemas may be published together.

4.1.3 URI Syntax for Change Schema Target Namespace

Each domain update change schema has a unique `targetNamespace` as its unique identifier. A `targetNamespace` for a domain update change schema includes a URL authority (Internet domain name) for a publication area in its base URI. The NIEM Publication Area URL is “`http://publication.niem.gov/`”. This is the base URI of the `targetNamespace` for all domain update schemas published in the NIEM Publication Area.

Rule 4-11: The `targetNamespace` for a domain update change schema **MUST** be a valid URL that adheres to the syntax defined by the following ABNF [RFC5234] grammar:

```
url = "http://" publicationDomain "/" domainPath "/"
      domainVersionNumber "/" domainUpdateVersionNumber "/"

publicationDomain = "publication.niem.gov" /
PUBLICATIONAREA

domainPath = "niem/domains/" domainName

domainName = "cbrn" / "emergencyManagement" /
"familyServices" / "immigration" /
"infrastructureProtection" / "intelligence" /
"internationalTrade" / "jxdm" / "maritime" / "screening" /
NEWDOMAIN

domainVersionNumber = 1*DIGIT *1("." 1*DIGIT *1("."
1*DIGIT))

domainUpdateVersionNumber = 1*DIGIT *1("." 1*DIGIT)
*1(("alpha" / "beta" / "rc" / "rev") 1*DIGIT)
```

In this grammar `NEWDOMAIN` is defined as a name for a new domain approved by the NIEM PMO. Also, `NEWPUBLICATIONAREA` is defined as the Internet domain name or authority of a different (local domain) publication area.

Note that version numbers adhere to [NIEM-MPD], and therefore, a final “/” is NOT allowed.

A domain may be created with permission from NBAC/NTAC through the domain update process. In this situation, the domain version number that is being updated is 0.

Each example `targetNamespace` below adheres to the rule above (the last example illustrates a typical new domain, in this called “foo”):

```
http://publication.niem.gov/niem/domains/cbrn/2.1/1/
http://publication.niem.gov/niem/domains/jxdm/4.1/2.3.1/
http://publication.niem.gov/niem/domains/foo/0/1/
```

The file name of a domain update change schema must adhere to the following rule:

Rule 4-12: A domain update change schema **MUST** bear the same file name as the reference schema in the NIEM release to which it applies.

This rule ensures that it is easy to quickly identify update change schemas associated with a particular domain.

4.1.4 Other XML Schema Artifacts

The [NIEM-MPD] specification allows several different types of XML schema artifacts in MPDs. This specification further restricts what a domain update may include. Domain updates contain authoritative content for use within IEPDs and EIEMs. They do not implement data exchanges.

Rule 4-13: A domain update package **MUST NOT** include a subset schema, extension schema, exchange schema, or constraint schema.

Note that this rule does not preclude a domain update from extending existing NIEM content. However, extension schemas are strictly IEPD/EIEM artifacts, and their conformance rules are more relaxed than are those for reference schemas. Domain update schemas must be reference schemas which contain authoritative data definitions that all IEPDs may use.

4.2 Catalog

[NIEM-MPD] **Rule 4-1** requires that an MPD include a catalog of metadata and artifact identifiers. This rule specifies the name, location, and content of the catalog. It also defines the `targetNamespace` for the catalog file definition (`mpd-catalog.xsd`) as: “`http://niem.gov/niem/mpd-catalog/1.0/`”. As an MPD, a domain update must adhere to this rule.

In addition to this rule, this specification defines further requirements on the URI syntax of the domain update; the nature and purpose of the change schema, catalog, change log, quality assurance report, and sample instance; and the lineage of the domain update.

4.2.1 URI Syntax for MPD

[NIEM-MPD] **Rules 4-4 and 4-5** require that a MPD have a valid URI and that the URI suffix be the MPD version number. Beyond these rules, [NIEM-MPD] leaves MPD URI syntax to the developer. However, this specification restricts URI syntax for a domain update as follows:

Rule 4-14: The URI of a domain update **MUST** be a valid URL that adheres to the syntax defined by the following ABNF [RFC5234] grammar:

```
url = "http://" publicationDomain "/" domainUpdatePath "/"
      domainVersionNumber "/" domainUpdateVersionNumber "/"

publicationDomain = "publication.niem.gov" /
NEWPUBLICATIONAREA
```

```

domainUpdatePath = "niem/update/" domainName

domainName = "cbrn" / "emergencyManagement" /
"familyServices" / "immigration" /
"infrastructureProtection" / "intelligence" /
"internationalTrade" / "jxdm" / "maritime" / "screening" /
NEWDOMAIN

domainVersionNumber = 1*DIGIT *1("." 1*DIGIT *1("."
1*DIGIT))

domainUpdateVersionNumber = 1*DIGIT *1("." 1*DIGIT)
*1(("alpha" / "beta" / "rc" / "rev") 1*DIGIT)

```

In this grammar NEWDOMAIN is defined as a name for a new domain approved by the NIEM PMO. Also, NEWPUBLICATIONAREA is defined as the Internet domain name or authority of a different (local domain) publication area.

The following example URIs adhere to the rule above (the last example illustrates a typical new domain, in this case a domain called “foo”):

```

http://publication.niem.gov/niem/update/cbrn/2.1/1/
http://publication.niem.gov/niem/update/jxdm/4.1/2.3.1/
http://publication.niem.gov/niem/update/foo/0/1/

```

A coordinated domain update contains multiple URIs that could be used for the domain update package. Any of those URIs can be used for the URI in a coordinated domain update because it uniquely identifies the domain update package.

4.2.2 Artifact Nature and Purpose

[NIEM-MPD] **Rule 4-6** requires that every MPD artifact have a corresponding File element in its catalog. [NIEM-MPD] **Catalog Schema** further requires that a File entry have both “natureURI” and “purposeURI” attributes. [NIEM-MPD] enumerates the possible values for these attributes in a Web Ontology Language (OWL) file.

For a domain update, this specification requires that the natureURI and the purposeURI attributes have values as defined by the following rules:

Rule 4-15: The File entry for an incremental change schema **MUST** have a “natureURI” value = “http://niem.gov/niem/resource/mpd-lexicon/1.0/nature#xsd” and a “purposeURI” value = “http://niem.gov/niem/resource/mpd-lexicon/1.0/purpose#incremental-schema”.

Rule 4-16: The File entry for a replacement change schema **MUST** have a “natureURI” value = “http://niem.gov/niem/resource/mpd-lexicon/1.0/nature#xsd” and a “purposeURI” value =

```
"http://niem.gov/niem/resource/mpd-lexicon/1.0/purpose#replacement-schema".
```

Rule 4-17: The File entry for a catalog **MUST** have a "natureURI" value = "http://niem.gov/niem/resource/mpd-lexicon/1.0/nature#xml" and a "purposeURI" value = "http://niem.gov/niem/resources/mpd-lexicon/1.0/purpose#catalog".

Rule 4-18: The File entry for a change log **MUST** have a "natureURI" value = " http://niem.gov/niem/resource/mpd-lexicon/1.0/nature#xml" and a "purposeURI" value = "http://niem.gov/niem/resources/mpd-lexicon/1.0/purpose#changelog".

Rule 4-19: The File entry for a sample instance **MUST** have a "natureURI" value = "http://niem.gov/niem/resource/mpd-lexicon/1.0/nature#xml" and a "purposeURI" value = "http://niem.gov/niem/resources/mpd-lexicon/1.0/purposes#sample-instance".

Rule 4-20: The File entry for a quality assurance report **MUST** have a "natureURI" value = " http://niem.gov/niem/resource/mpd-lexicon/1.0/nature#xls" and a "purposeURI" value = "http://niem.gov/niem/resources/mpd-lexicon/1.0/purpose#quality-assurance-report".

4.2.3 Lineage

[NIEM-MPD] defines an optional "Relationship" element for the catalog to describe the lineage of a MPD. For a domain update, this specification requires that the catalog include a "Relationship" element to indicate the version of this specification to which the domain update conforms, the release upon which the domain update is based, and the domain update upon which the domain update is based, if any.

Rule 4-21: The catalog for a domain update package **MUST** include a "Relationship" element whose "relationshipCode" attribute has the value

"conforms_to" and whose "resourceURI" attribute has the value "http://niem.gov/niem/specification/domain-update/1.0/".

Rule 4-22: The catalog for a domain update package MUST include a "Relationship" element whose "relationshipCode" attribute has the value "updates" and whose "resourceURI" attribute has the value of the URI of the release upon which the domain update is based.

The [NIEM-MPD] defines a NIEM release as a MPD and therefore NIEM releases will have a declared URI. For NIEM releases that predate [NIEM-MPD] use the following:

For NIEM 2.1 use "http://niem.gov/niem/2.1/"

For NIEM 2.0 use "http://niem.gov/niem/2.0/"

For NIEM 1.0 use "http://niem.gov/niem/1.0/"

Rule 4-23: The catalog for a domain update package MUST include a "Relationship" element whose "relationshipCode" attribute has the value "updates" and whose "resourceURI" attribute has the value of the URI of the domain update upon which the domain update is based.

While it is possible to derive the domain and domain version from the domain update URI, it is not always possible to derive the release to which the domain update applies, since it is possible for the same version of a domain to appear in multiple releases. As a result, it is necessary for the domain update to explicitly indicate the release to which it applies.

4.3 Change Log

[NIEM-MPD] defines a change log as a required artifact that tracks changes from one version of a MPD to another. It mandates that exactly one change log exist in a MPD. Additional constraints exist on a domain update change log.

Rule 4-24: A domain update change log MUST NOT define changes in any files other than those contained in the domain update package within which it resides.

The change log provides tools with the documentation and traceability needed to facilitate automation and tool support for domain updates and integration into future NIEM releases. To facilitate precise tool identification and processing of all data in a domain update, the change log must identify every change to content in the schemas.

Rule 4-25: A domain update change log MUST identify every change to domain data components represented in the change schema artifacts, including delete transactions.

Deletions do not explicitly appear in change schemas. Therefore, the rule above highlights the requirement that the change log must identify all changes including deletions.

A coordinated domain update will have a single change log artifact identifying all changes represented by all change schemas of all participating domains.

4.4 Quality Assurance Reports

In order to allow a domain steward to quickly respond to the user community's needs and adjust domain content on an independent timeline, some parts of the NIEM quality control process will be deferred to the regular release development cycle. Therefore, NBAC review and cross-domain and/or Core harmonization will occur when the content of a domain update is being integrated into the next NIEM release. This will help to streamline domain update publication. Reduction of bottlenecks will become increasingly important as NIEM grows.

Increased flexibility to independently adjust content while preserving high standards of NIEM quality, make it necessary to enforce a baseline level of quality control that is consistent and automated. Quality tests and conformance requirements are defined across a number of specifications, including this specification: the [NIEM-MPD], the [NIEM-NDR], and the NIEM *Quality Assurance Strategy and Plan (QASP)* [NIEM-QASP]. The expectation of complete mastery and retention of each of these specifications is a burden to place on a domain steward, and even if achieved, rules may still be unintentionally misapplied or overlooked. Therefore, a *reference domain update quality assurance test suite* is provided for assistance. This test suite runs automated quality control tests from all relevant specifications and provides guidance information for tests that are not yet or cannot be automated.

Rule 4-26: A domain update MUST be tested by the reference domain update quality assurance test suite before it can be published.

Tests from the reference test suite will be aggregated into four categories: conformance, quality assurance, impact, and harmonization. Conformance tests will check the domain update against NIEM, MPD, and domain update-specific rules. Quality tests will provide additional checks to help ensure consistency, good modeling practices, and to identify potential problems for additional domain steward review.

Rule 4-27: A domain update MUST pass all required conformance and quality assurance tests provided by the reference domain update quality assurance test suite.

Impact tests will compare the submitted domain update with other published domain updates and other domains to determine the effect of domain update changes on each domain.

Harmonization tests will identify potential areas of overlap or duplication with other domains and/or Core. Because a domain update does not require cross-domain harmonization, it is not required to pass these tests for publication. However, by reviewing the results of these tests all domain stewards to better understand (1) the effects of an update, (2) some of the potential work necessary during the next major or minor release, and (3) how to change content in ways that will result in smoother integration in the future.

Rule 4-28: The artifact(s) produced as a result of testing a domain update against the reference domain update quality control test suite **MUST** be included as artifacts in the domain update on which they were processed.

The quality assurance reports represent the current state of a domain update, as well as the quality tests that were available, at the time they were run on an update. Harmonization and impact reports may change if rerun after other domains have made modifications through domain updates. Furthermore, domain updates can fail previously passed conformance and quality assurance tests when rerun at a later date as new rules are added to the testing tools. This can also occur because of NIEM NDR changes that are reflected in conformance and quality assurance tools. In such cases, no changes to the domain update will be required, but before integration into a NIEM release, content will have to be modified to ensure conformance to new rules.

4.5 Sample Instances

[NIEM-MPD] indicates that sample XML instances can (in some cases, must) be included in a MPD. In the case of a domain update, the purpose of a sample instance is to illustrate the use of the components defined in its change schema. Sample instances can assist developers to use domain components as they were intended by providing authoritative use cases.

Rule 4-29: A domain update package **MAY** optionally include sample instances to demonstrate example uses of the data components defined in its change schema.

Rule 4-30: A sample instance in a domain update package **SHOULD** validate against the change schema included in the package along with any imported schemas.

Having a sample instance that does not validate against the change schema in a domain update package does not make the domain update package non-conformant, but would be a very bad practice as the sample instance would fail to provide a useful sample.

Appendix A: Publication Area

The [NIEM-HLVA] defines a *publication area*, a persistent location for domains to publish domain updates. A publication area serves two major purposes:

1. Allowing updates to be published and utilized between normal NIEM release cycles.
2. Providing a platform and location for staging changes for domain reconciliation and Core synchronization prior to NIEM releases.

This section is normative and shows the impact of a domain update package on a publication area. In doing so, it specifies behavior of a publication area and the lifecycle of files once they have been published. The interfaces for interacting with a publication area and additional rules that govern interactions between multiple publication areas will be defined in a future version of this specification.

A.1 Publication Area Name

Publication areas must be identifiable for reference by tools and also to enable the download of published updates. A publication area uses its Internet domain name or authority as its unique identifier.

Rule A-1: A publication area must have a URI that is "http://" followed by its Internet domain name or authority.

A.2 Publication Area Content

Publication area content consists of artifacts from domain updates that a domain steward provides to the publication area. An artifact is considered to be published once it is persistently available at a specific URL. Publication area content may include the domain update package, its catalog, its sample instances, or its quality assurance report, but in order to be of use to IEPD developers, the publication area must publish the change schemas and change log.

Rule A-2: A publication area **MUST** publish all change schemas and the change log from the domain update package.

The publication area uses the catalog file to identify the domain update artifacts and to gather metadata about the update. Subsequently the artifacts are made available within the publication area. The domain update package itself may be transient; once the domain update package has been delivered to a publication area, one cannot presume the update package can ever be retrieved in its original form. A publication area may provide a feature to retrieve the original domain update package, but that feature is not a requirement of this specification.

A.2.1 Change Schemas

A publication area must publish the change schema files to a URL derived from the change schema's `targetNamespace` and `filename`.

Rule A-3: A change schema from a domain update **MUST** be published in a publication area to the URL specified by the `targetNamespace` of the change schemas, followed by a "/" character, followed by the filename of the change schema.

For example, for a single change schema with `targetNamespace`,

```
http://publication.niem.gov/niem/domains/jxdm/4.1/2.3.1/
```

and file name, `jxdm.xsd`, the publication area must publish the change schema to the following URL:

```
http://publication.niem.gov/niem/domains/jxdm/4.1/2.3.1/jxdm.xsd
```

After publication, IEPD and EIEM developers may use the change schemas in IEPDs and EIEMs. As such, the change schema must remain available and unchanged to ensure those MPDs remain valid.

Rule A-4: A publication area **SHOULD NOT** modify, remove, or move a change schema to a new URL after publishing it.

There are rare instances where a schema must be removed after it has been published to a publication area. This could include a security violation or a directive from the NIEM PMO. In the majority of cases however, the change schema should never be moved or modified.

A.2.2 Change Logs

A publication area must publish the change log to a URL derived from the associated change schema `targetNamespace`.

Rule A-5: The change log from a domain update **MUST** be published to the URL specified by the `targetNamespace` of the associated change schema, followed by a "/" character, followed by `changelog.xml`.

For example, for the change log associated with a change schema whose `targetNamespace` is

```
http://publication.niem.gov/niem/domains/jxdm/4.1/2.3.1/
```

The publication area must publish the change log to the following URL:

```
http://publication.niem.gov/niem/domains/jxdm/4.1/2.3.1/changelog.xml
```

Similar to the change schema, the change log should (in most cases) never be removed or modified.

Rule A-6: A publication area **SHOULD NOT** modify or remove the change log after it has been published.

These rules are also applicable for coordinated domain updates.

A.2.3 Domain Update Packages

While the change schema and change log from a domain update must be published in a publication area, the domain update package is not required to be available. However, if the domain update package is published within a publication area, it must be available at a URL derived from the domain update URI.

Rule A-7: A publication area MAY publish the domain update package at a downloadable location.

Rule A-8: If a publication area publishes a complete domain update package, then a URL specified by the URI of the domain update package, followed by “/update.zip” MUST resolve to that package.

For example, a publication area would publish a domain update package with the URI

```
http://publication.niem.gov/niem/update/jxdm/4.1/2.3.1/
```

To the following URL:

```
http://publication.niem.gov/niem/update/jxdm/4.1/2.3.1/update.zip
```

A.2.4 Coordinated Domain Updates

A coordinated domain update will contain more than one change schema, unless one of the domain changes contains only deletions in an incremental update. Nonetheless, a coordinated update always contains one and only one change log. The change log is associated with all change schemas, and consistent with Rule A-5, must be published in a single location for each change schema.

Rule A-9: A publication area that publishes a coordinated domain update MUST create a copy of the change log in the directory of each domain schema file(s).

Consider the example of a coordinated domain update that contains two change schemas, each of which has one of the following `targetNamespace` attributes:

```
http://publication.niem.gov/niem/domains/jxdm/4.1/2.3.1/  
http://publication.niem.gov/niem/domains/cbrn/2.1/1/
```

The publication area should publish a copy of the change log associated with the change schemas to the following URLs:

```
http://publication.niem.gov/niem/domains/jxdm/4.1/2.3.1/changelog.xml  
http://publication.niem.gov/niem/domains/cbrn/2.1/1/changelog.xml
```

Before a coordinated domain update can be published, each domain data steward for the domains involved in the update must have authorized the domain update.

Rule A-10: Each participating domain steward must authorize a coordinated domain update before it can be published in a publication area.

The exact mechanism for this will be defined in a later version of this specification.

A.2.5 Prerelease Domain Updates

Per [NIEM-MPD], a domain may indicate a prerelease domain update by a version number with the suffixes "alpha", "beta", or "rc"; where "alpha" indicates early development, "beta" indicates late development, and "rc" indicates release candidate. While prerelease domain updates and published domain updates are similar in content, prerelease domain updates may be incomplete, may contain errors, and may not be checked for quality. They are available for evaluation only. All domain updates are considered revisions, therefore the "rev" suffix described in [NIEM-MPD] is also not permitted.

Rule A-11: A publication area MUST NOT contain prerelease domain updates; specifically, a domain update with a version number that does not adhere to the following regular expression: $([0-9]+)(\.[0-9]+)^*$.

A domain steward must use other locations to facilitate review and testing of prerelease domain updates. Prior to publishing an operational version of a domain update, a domain steward should remove all preceding prerelease domain updates from public access.

A.3 Publication Area Service Interface

The official public NIEM Publication Area is:

`http://publication.niem.gov/`

The NIEM Publication Area is the default publication area for most domains and users, unless security or governance restrictions prevent a domain from using it.

Users may download and view updates located in the NIEM Publication Area using any standard Web browser. Alternative methods for accessing files in the NIEM Publication Area through the usage of APIs will be defined in the future.

Full details on usage of the NIEM Publication Area can be found in the Domain steward Technical Guide [NIEM-DSTG].

A.4 Multiple Publication Areas

This specification is designed to allow the existence of multiple publication areas. The official NIEM Publication Area is intended to be used in the majority of cases, but a domain may also create and host its own publication area as required by its organizational security or governance policies.

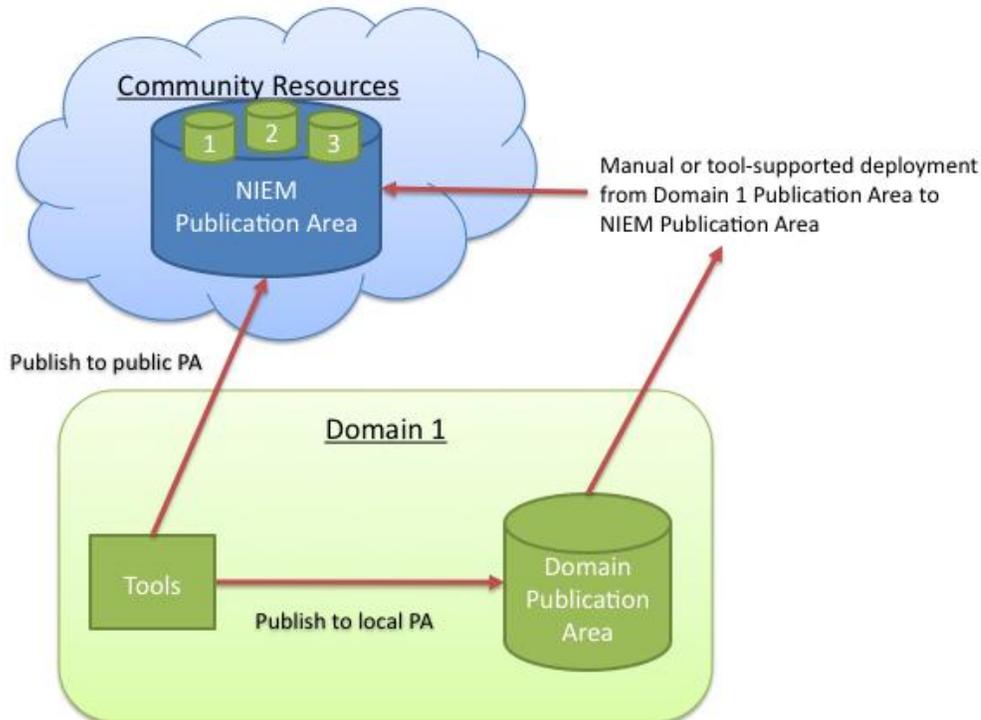


Figure A-1 - Interaction between Publication Areas

This specification currently does not define the APIs necessary for communication between multiple publication areas, or the mechanisms to allow caching of domain updates by the local publication areas from the NIEM Publication Area. These will be defined in a future version of this specification.

A.5 Future Versions

The current version of this specification deliberately does not specify a few key concepts. These will be determined through limited beta tests and community feedback. One concept to be defined in the future is a deprecation change log format.

A.5.1 Application Program Interface (API)

APIs for several types of use cases will be specified in the future: (1) Interaction between multiple publication areas; (2) Programmatic access for tool developers.

Potential operations that might be enabled by APIs in the future include:

- `get_schema(schema_uri)`
- `get_changelog(changelog_uri)`

- `get_deprecationlog(deprecationlog_uri)`
- `publish_domain_update(domain_update)`
- `deprecate_domain_update(domain_update_uri)`

Appendix B: Sample Schemas for Incremental Update and Replacement Update

The following example illustrates a domain update. A domain currently has three components: `PersonType`, `PersonName`, and `Address`. The domain update creates the new component `Alias`, and modifies the existing component `PersonType`.

B.1 Incremental Schema Domain Update

The incremental schema domain update includes both new and modified components in the change schema. In the example above, the incremental domain update would contain two components: `Alias` and the modified `PersonType`.

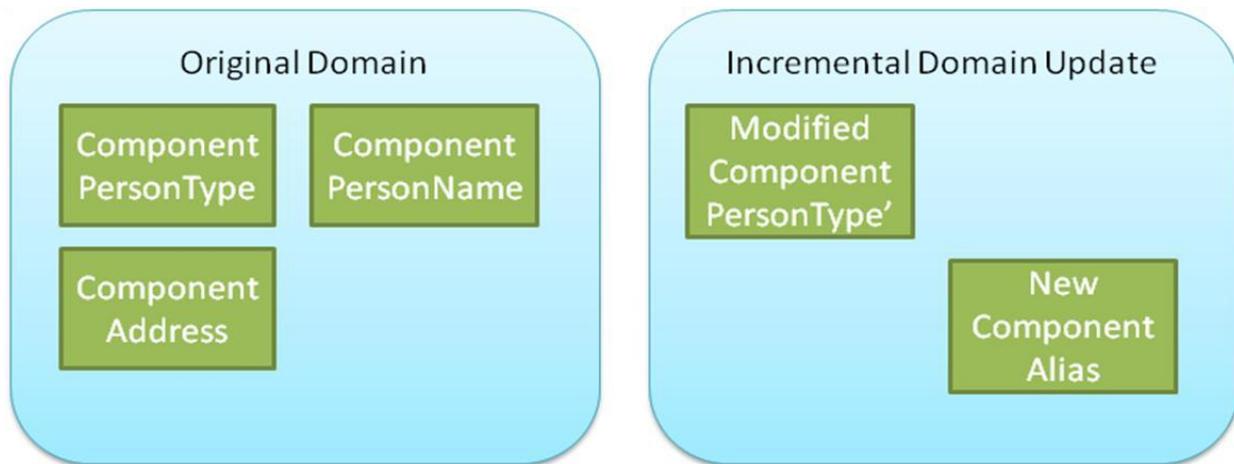


Figure B-1 - Incremental Domain Update

The listing below illustrates the XML schema for this incremental domain update.

Listing B.1

```
<xsd:schema
  xmlns:s = "http://niem.gov/niem/structures/2.0"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns:nc = "http://niem.gov/niem/niem-core/2.0"
  xmlns:i = "http://niem.gov/niem/appinfo/2.0"
  xmlns:exd = "http://publication.niem.gov/niem/domains/example-
domain/2.1/1/"
  xmlns:ex = "http://niem.gov/niem/domains/example-domain/2.1"
  targetNamespace="http://publication.niem.gov/niem/domains/example-
domain/2.1/1/"
  version="1">
  <xsd:annotation>
    <xsd:appinfo>
```

```

    <i:ConformantIndicator>true</i:ConformantIndicator>
  </xsd:appinfo>

</xsd:annotation>

<xsd:import
  namespace="http://niem.gov/niem/structures/2.0"
  schemaLocation="../../../structures/2.0/structures.xsd"/>
<xsd:import
  namespace="http://niem.gov/niem/appinfo/2.0"
  schemaLocation="../../../appinfo/2.0/appinfo.xsd"/>
<xsd:import
  namespace="http://niem.gov/niem/niem-core/2.0"
  schemaLocation="../../../niem-core/2.0/niem-core.xsd"/>
<xsd:import
  namespace="http://niem.gov/niem/domains/example-domain/2.1/"
  schemaLocation="../../../domains/example-domain/2.1/example-
domain.xsd"/>

<xsd:complexType name="PersonType">
  <xsd:annotation>
    <xsd:appinfo>
      <i:Base i:namespace="http://niem.gov/niem/structures/2.0"
        i:name="Object"/>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="s:ComplexObjectType">
      <xsd:sequence>
        <xsd:element ref="ex:PersonName" minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element ref="exd:PersonAlias" minOccurs="0"
          maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

  <xsd:element name="Person" type="exd:PersonType" nillable="true"/>
  <xsd:element name="PersonAlias" type="nc:TextType" nillable="true"/>
</xsd:schema>

```

B.2 Replacement Schema Domain Update

The replacement schema domain update includes every component in the domain. In the above example, the replacement domain update would contain four components: the modified `PersonType`, the new components `Alias`, and the unchanged components `PersonName` and `Address`. This is illustrated in Figure B-2. This is a complete replacement of the components in the domain model, as compared to the incremental schema.

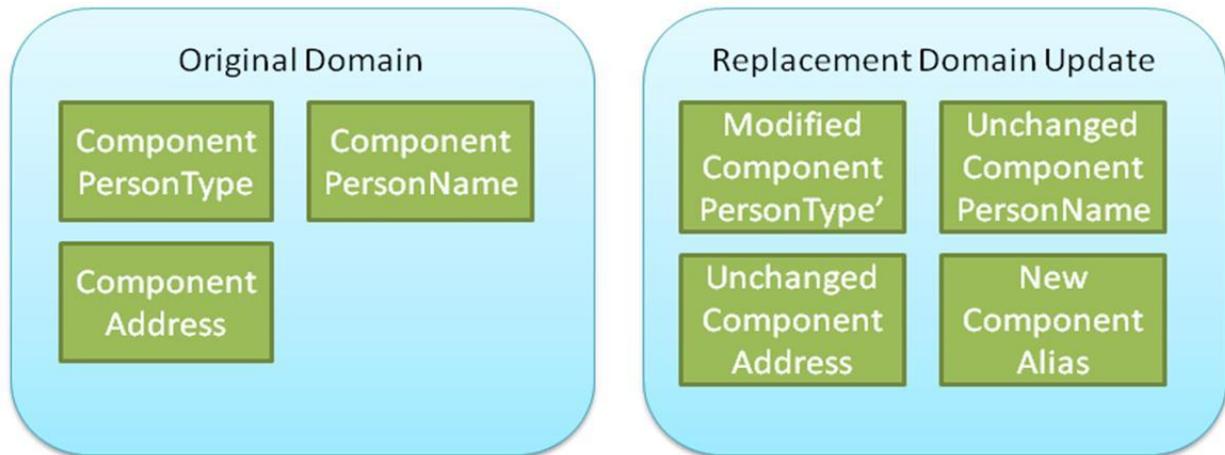


Figure B-2 - Replacement Domain Update

The listing below illustrates an XML schema for this replacement domain update.

Listing B.2

```
<xsd:schema
  xmlns:s = "http://niem.gov/niem/structures/2.0"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns:nc = "http://niem.gov/niem/niem-core/2.0"
  xmlns:i = "http://niem.gov/niem/appinfo/2.0"
  xmlns:ex = "http://niem.gov/niem/domains/example-domain/2.1/1/"
  targetNamespace = "http://publication.niem.gov/niem/domains/
example-domain/2.1/1/"
  version="1">

  <xsd:annotation>
    <xsd:appinfo>
      <i:ConformantIndicator>true</i:ConformantIndicator>
    </xsd:appinfo>
  </xsd:annotation>

  <xsd:import
    namespace="http://niem.gov/niem/structures/2.0"
    schemaLocation="../../../structures/2.0/structures.xsd"/>
  <xsd:import
    namespace="http://niem.gov/niem/appinfo/2.0"
    schemaLocation="../../../appinfo/2.0/appinfo.xsd"/>
  <xsd:import
    namespace="http://niem.gov/niem/niem-core/2.0"
    schemaLocation="../../../niem-core/2.0/niem-core.xsd"/>

  <xsd:complexType name="PersonType">
    <xsd:annotation>
      <xsd:appinfo>
        <i:Base i:namespace="http://niem.gov/niem/structures/2.0"
          i:name="Object"/>
      </xsd:appinfo>
    </xsd:annotation>
  </xsd:complexType>
</xsd:schema>
```

```
</xsd:appinfo>
</xsd:annotation>
<xsd:complexContent>
  <xsd:extension base="s:ComplexObjectType">
    <xsd:sequence>
      <xsd:element ref="ex:PersonName" minOccurs="0"
        maxOccurs="unbounded"/>
      <xsd:element ref="ex:PersonAlias" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:element name="Address" type="nc:TextType" nillable="true"/>
<xsd:element name="Person" type="ex:PersonType" nillable="true"/>
<xsd:element name="PersonAlias" type="nc:TextType" nillable="true"/>
<xsd:element name="PersonName" type="nc:TextType" nillable="true"/>
</xsd:schema>
```

Appendix C: Glossary

Catalog File – Describes all of the files and associated metadata within a domain update

Change Log – XML representation of all the changes made to the schema files of a domain update

Change Request – A formal request for new content or modifications to be incorporated into a NIEM release.

Change Schema – Reference schema or schema set that contain the content for a MPD

Coordinated Domain Update – A domain update where multiple domains work together to publish a closely interdependent domain update as a single domain update package

Data Component – An XML element, attribute, or type definition in NIEM.

Domain Steward – The individual responsible for the content and model of a domain

Domain Update - An artifact that allows a domain steward to submit changes for their domain content for immediate use by developers.

Domain Update Change Schema – A NIEM reference schema that defines new data components or changes to existing domain data components in a domain update.

Domain Update Package – A complete domain update MPD.

Enterprise Information Exchange Model (EIEM) – An MPD that incorporates Business Information Exchange Components using NIEM.

Harmonization – A process for modeling and integrating new and existing data components in ways that remove duplication, resolve conflicts, reduce variation, and achieve consistency. The goal of harmonization is to bring new content into NIEM while reestablishing and maintaining standardization and uniformity across the data model under the NIEM NDR.

Incremental Domain Update – A domain update where the only components in the schema are ones that have been modified or added

Information Exchange Model (IEM) – One or more NIEM-conforming XML schemas that together specify the structure, semantics, and relationships of XML objects. These objects are consistent XML representations of information. Currently, four IEM classes exist in NIEM: (1) numbered release, (2) domain update, (3) Information Exchange Package Documentation (IEPD), and (4) Enterprise Information Exchange Model (EIEM).

Information Exchange Package Description (IEPD) – Definition for a recurring data exchange that consists of a minimal but complete set of artifacts to describe a NIEM information exchange.

Major Release – A NIEM release that includes updates and synchronization to NIEM Core and the domains.

Micro Release – A NIEM release generated at the request of a domain at any time with approval from the NIEM PMO and incorporates all coherent domain updates into the release.

Minor Release – A NIEM release that involves harmonization of domains. No harmonization of NIEM Core takes place.

Model Package Description (MPD) – A compressed archive of files that contains one and only one of the four classes of NIEM IEM, as well as supporting documentation and other artifacts. A MPD is self-documenting and provides sufficient normative and non-normative information to allow technical personnel to understand how to use and/or implement the IEM it contains.

NIEM Release – A high quality, coherent set of all schemas that make up NIEM that is independent of all other releases.

NIEM Release Manager – The entity responsible for publishing and ensuring the quality of NIEM major or minor releases.

Publication Area – A persistent location to which domain update artifacts are published.

Quality Assurance Report – Report generated from automated checks that assist the developer in meeting NDR rules.

Reference Schemas – Schemas intended to be the authoritative definition schema for a NIEM namespace.

Replacement Domain Update – A domain update where component additions, modifications, and deletions are integrated with the domain components that are not changing in the update schema.

Appendix D: **Acronyms and Abbreviations**

API – Application Programming Interface

DSTG - Domain steward Technical Guide

EIEM – Enterprise Information Exchange Model

HLVA - High Level Version Architecture

IEM – Information Exchange Model

IEPD – Information Exchange Package Description

MPD – Model Package Description

NBAC - NIEM Business Architecture Committee

NCCT - NIEM Configuration Control Tool

NDR - Naming and Design Rules

NIEM – National Information Exchange Model

NTAC - NIEM Technical Architecture Committee

PMO – Project Management Office

QASP - Quality Assurance Strategy and Plan

URI - Uniform Resource Identifier

URL – Uniform Resource Locator

W3C – World Wide Web Consortium

XML – eXtensible Markup Language

XSD - XML Schema Definition

Appendix E: References

[NIEM-DSTG]: NIEM Domain Steward Technical Guide, NIEM Technical Architecture Committee (NTAC). When approved will be available from

<http://reference.niem.gov/niem/guidance/domain-steward-technical-guide/1.0/>

[NIEM-MPD]: NIEM Model Package Description (MPD) Specification, Version 1.0, NIEM Technical Architecture Committee (NTAC). When approved will be available from

<http://reference.niem.gov/niem/specification/model-package-description/1.0/>

[NIEM-QASP]: NIEM Quality Assurance Strategy and Plan, Version 1.0, NIEM Technical Architecture Committee (NTAC), 20 May 2008.

Available from: <http://reference.niem.gov/niem/guide/qa-strategy-and-plan/1.0/>

[NIEM-HLVA]: NIEM High Level Version Architecture (HLVA), Version 1.0, NIEM Technical Architecture Committee (NTAC), 2008.

Available from <http://reference.niem.gov/niem/specification/high-level-version-architecture/1.0/>

[NIEM-NDR]: NIEM Naming and Design Rules (NDR), Version 1.3, NIEM Technical Architecture Committee (NTAC), 31 October 2008.

Available from <http://reference.niem.gov/niem/specification/naming-and-design-rules/1.3/>

[RFC2119]: Bradner, S. Key words for use in RFCs to Indicate Requirement Levels, IETF RFC 2119, March 1997. Available from <http://www.ietf.org/rfc/rfc2119.txt>

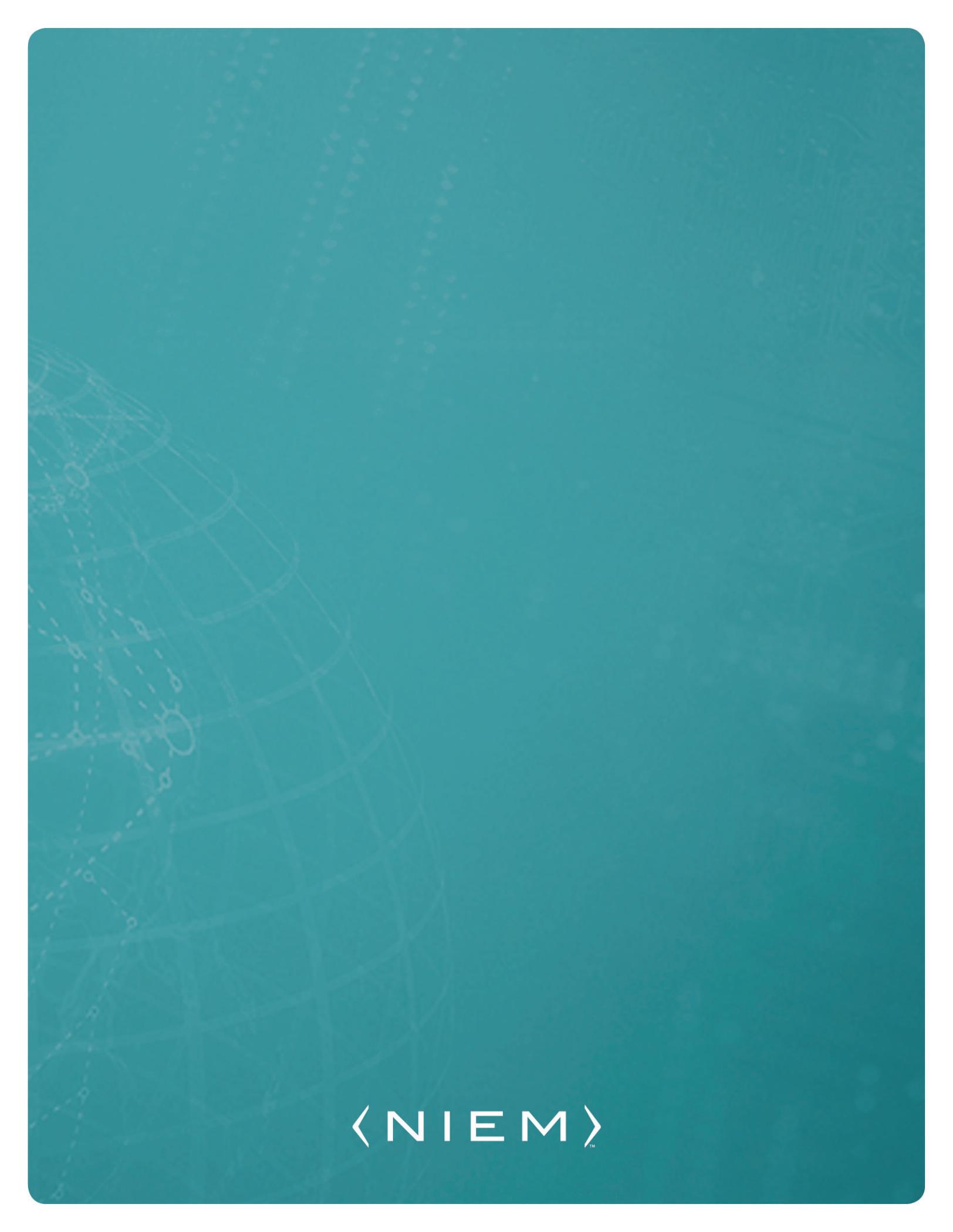
[RFC5234]: Crocker, D. and Overell, P. Augmented BNF for Syntax Specifications: ABNF, IETF RFC 5234, January 2008. Available from <http://www.ietf.org/rfc/rfc5234.txt>

[NIEM-CRT]: NIEM Change Request Template (CRT), Version 1.0, NIEM Technical Architecture Committee (NTAC). Available from

<http://reference.niem.gov/niem/resource/change-request-template/1.0/>

[NIEM-UML]: UML Profile for NIEM, Version 1.0, NIEM Technical Architecture Committee (NTAC). Available from

<http://reference.niem.gov/niem/specification/uml-profile/1.0/>



< NIEM >