

National Information Exchange Model — High-Level Version Architecture

Version 5.0

July 26, 2021

NIEM Technical Architecture Committee (NTAC)

URI

<http://reference.niem.gov/niem/specification/high-level-version-architecture/5.0/>

Contents

URI.....	1
Contents.....	1
Abstract.....	3
Status	3
1. Introduction	3
1.1. Scope.....	3
1.2. Background	4
2. Objectives.....	5
3. Overview	5
3.1. NIEM Release Artifacts.....	6
3.2. Data Stores.....	8
3.3. Activities.....	8
3.4. A Walkthrough of Versioning.....	9

4. Activities.....	11
4.1. Domain Reconciliation	11
4.2. Cross-Domain Harmonization	15
4.3. Core Synchronization	16
5. NIEM Releases.....	16
5.1. Properties of a NIEM Release	16
5.2. Status of a NIEM Release	17
5.3. Version Identifiers for NIEM Releases	17
6. Data Stores.....	17
6.1. The Release Area.....	18
6.2. The Publication Area	18
6.3. The Issue Tracking Area	20
7. Additional Artifacts	21
7.1. Change Log.....	21
7.2. Namespace URI Directive.....	22
8. General Release Timelines	23
8.1. Release Cycle Stages	24
8.2. Release Cycle Exceptions and Options.....	25
8.3. Outline of the Three-Year Cycle.....	26
9. Summary	27
Appendix A. Acronyms and Abbreviations.....	29
Appendix B. References	30
Appendix C. Index of Definitions.....	32
Appendix D. Revision History	33

Abstract

This document describes the National Information Exchange Model (NIEM) version architecture. It identifies the processes, artifacts, and responsibilities required to produce new releases of the NIEM model. It also establishes a regular cycle for predictable and manageable NIEM updates.

Status

This document presents a version architecture that has evolved through the collaborative work of the NIEM Technical Architecture Committee (NTAC) and the NIEM Business Architecture Committee (NBAC).

This document is a product of the NIEM Management Office (NMO).

Updates, revisions, and errata for this specification are posted to <https://github.com/NIEM/NIEM-HLVA-Spec>. Please submit comments on this specification as issues at <https://github.com/NIEM/NIEM-HLVA-Spec/issues>.

1. Introduction

This document describes the National Information Exchange Model (NIEM) version architecture at a high level. It describes the approach and the major artifacts involved, without specifying them at a detailed level.

The version architecture discusses issues that have inherently conflicting tradeoffs. It seeks a balance of concepts drawn from previous NIEM Technical Architecture Committee (NTAC) discussions and ideas. Major tradeoffs include:

- Timeliness versus stability
- Domain autonomy versus interoperability
- Ease for information exchange package documentation (IEPD)¹ developers versus ease for domain developers

1.1. Scope

This High Level Version Architecture (HLVA) describes how NIEM governance updates the data components and schema documents that comprise a NIEM release.

¹ An IEPD is a collection of NIEM artifacts that define and describe the context, content, semantics, and structure of one or more implementable information exchanges. (NIEM IEPD Specification)

This specification does not address version management for an Information Exchange Package Documentation (IEPD). For more information about IEPDs, see the IEPD Specification (<https://niem.github.io/MPD-Spec/v5.0/niem-iepd-spec.html>) and general IEPD information on niem.github.io (<https://niem.github.io/reference/iepd/>).

This document assumes that the reader is familiar with NIEM and its artifacts. Readers needing additional background on NIEM should refer to the NIEM GitHub site at <https://niem.github.io>.

1.2. Background

A NIEM release is a coherent set of XML schema documents and supporting artifacts representing a specific version of the NIEM data model. Implementers of IEPDs use components of the data model to define information exchange packages (IEP)² that will be transmitted in an implementation. The goal of NIEM is to define common data components that are highly reusable, to improve the economy of implementing information exchanges, to increase comprehension of those exchanges, and to reduce errors in exchanges.

The NTAC and the NIEM Business Architecture Committee (NBAC) govern NIEM architecture and content respectively. The NBAC and NTAC may appoint standing subcommittees and transitory tiger teams who take responsibility for specific issues. NTAC and NBAC make technical recommendations to the NMO. The NMO is the operational body that directs and executes NIEM daily operations, development, maintenance, and public relations. The Executive Steering Council (ESC) is the senior executive level group that reviews and approves major NIEM directions and activities.

[Definition: NIEM governance]

The aggregation of the Executive Steering Council (ESC), the NIEM Management Office (NMO), the NIEM Business Architecture Committee (NBAC), the NIEM Technical Architecture Committee (NTAC), and the NIEM Domains.

The data components that make up NIEM are published as a set of NIEM releases. Each NIEM release is composed of a set of schema documents that include a NIEM Core schema document, special infrastructure schema documents (`structures.xsd` and `appinfo.xsd`), multiple domain schema documents, and code table schema documents.

The NIEM content model is intentionally partitioned into specialized domains under NIEM Core.

[Definition: NIEM Core]

² An IEP is an actual NIEM message exchanged between stakeholders. The format for this message is defined by the schemas and artifacts in its corresponding Information Exchange Package Documentation (IEPD). (NIEM IEPD Specification)

The content of a NIEM release that is common or universal to all (or almost all) domains.

[Definition: NIEM domain]

A partition of a NIEM release that represents specialized data requirements for a particular community of interest or line of business. This term may also refer to the governance group for a domain.

A NIEM domain is sometimes associated with or managed by an organization independent of NIEM, and may have its own rules, procedures, schedule, and priorities, subject to the NIEM principles that require openness, balance, due process, an appeals process, and consensus. Each domain or code table schema document has a body of domain representatives that is responsible for its content and maintains its own timeline for updates and publication. For this document, the term domain applies to both domain and code table working groups.

For more information on what is included in a NIEM Core, what domains and code table schema documents are defined, and what they contain, visit the NIEM Release Artifacts page at <https://niem.github.io/reference/release/artifacts/>.

2. Objectives

The NIEM HLVA described by this document is meant to achieve the following objectives:

- A domain may issue a domain update as needed to publish content changes outside of the regular release cycle.
- IEPDs may reuse content from NIEM releases and domain updates.
- Domain updates are incorporated into the next NIEM release.
- A reliable schedule exists for releases and release activities.
- All NIEM content changes are visible, via uniquely identified schemas and from change logs.

3. Overview

The HLVA provides a framework for changes to be performed, in a systematic way, on a set of artifacts. These are outlined in subparts of this section. The activities that lead to changes in the schema documents and other artifacts are driven by a desire to improve the model. Improvements include correcting errors, meeting previously unforeseen requirements, and adapting to new needs and new uses of the model. One concern is how to define improvement to the model. A common term for improving systems is harmonization, a term that was adopted for this effort.

[Definition: harmonization]

Harmonization (of a schema document or set of schema documents) is the process of modifying a schema document set in an incremental fashion for the purpose of improving the quality of the schema document set with respect to some criteria. For example, a schema document set may be harmonized to reduce semantic overlap of its components. A schema document set may also be harmonized to ensure uniformity of the vocabulary used in definitions of its components.

Harmonization may be conducted within a single schema document, or may be conducted across multiple schema documents of a schema document set.

The version architecture does not anticipate that NIEM will ever be completely harmonized (a theoretically perfect state for a schema document set, in which all components perfectly satisfy all criteria). In practice, a fully harmonized state is never reached, and the goal of harmonization is incremental improvement, rather than perfection.

This section outlines the basic structure of the architecture. Many details are in later sections.

3.1. NIEM Release Artifacts

The HLVA defines various kinds of artifacts, including:

[Definition: NIEM release]

A NIEM release is a versioned set of schema documents published by the NIEM Management Office. Each schema document defines data components for use in information exchanges. A release should:

- be easy to use.
- define each concept in a coherent way.
- be independent of other releases, although any given schema document may occur in multiple releases.
- be of high quality, and has been vetted by NIEM governance.

A NIEM release can be any one of three types:

[Definition: major release]

A NIEM release in which the NIEM Core reference schema document has changed since the previous releases. A major release may include architectural changes, as well as content changes to any namespace. The first integer of the version number indicates the major release series and the second integer will always be “.0”; for example, NIEM versions 1.0, 2.0, and 3.0 are different major releases.

[Definition: minor release]

A NIEM release in which the NIEM Core has not changed from previous releases in the series, but at least one or more domain reference schema documents have changed. A second integer greater than zero in the version number indicates a minor release (for example, NIEM version 2.1). Note also that major version 2.0 and minor version 2.1 are in the same series (i.e., series 2) and contain the same NIEM Core schema document. A minor release will not include architecture changes.

As a general rule, a published NIEM release is never revoked or removed from the release area. However, updates to a NIEM release may be published.

[Definition: published update]

A published update is a schema document or set of schema documents that is issued by a NIEM domain or NIEM governance body. An update may be new content or may be an update of content that was previously included in a NIEM release. A published update may define new versions of content from NIEM release or other published content. The issuing body vets each update before publication, but the update is not subject to review by other NIEM bodies. A published update must be conformant, but otherwise it has fewer constraints on quality than does a NIEM release.

There are currently two kinds of published updates in NIEM:

[Definition: domain update]

A domain update adds or modifies components in its own schema documents associated with a previously published NIEM release. A domain update is published to the publication area by the NIEM domain owning the schema document that the domain update modifies.

[Definition: core supplement]

A special NIEM release that adds or supplements components in an existing (i.e., previously published) NIEM Core within a NIEM release. A core supplement is published to the release area by the NMO (at the recommendation of NBAC and NTAC). A core supplement is additive or supplemental in nature; it cannot replace or modify components in a published NIEM Core.

Through review and usage of published releases, potentially new or missing requirements, flaws, or other concerns will be identified in NIEM. These issues are recorded for review, discussion, and action (as needed) by NIEM governance.

[Definition: NIEM issue]

Users of NIEM schema documents and components will find problems that should be addressed. These will include erroneous syntax or definitions, inconsistencies between different parts of the model, and new requirements that are not yet satisfied. A NIEM issue is a record of one of these problems. An issue's record may include discussion of the problem, possible solutions, and changes to the model required to resolve the problem.

3.2. Data Stores

There are several data stores in the architecture. General terms are used here, to avoid introducing any requirements or restrictions associated with an existing tool, such as a registry, repository, or specific Web software such as Bugzilla or Sharepoint. This description seeks only to describe requirements and functionality for these data stores.

[Definition: release area]

A network location at which NIEM releases are stored and made publicly available. NIEM releases are now hosted on two sites: release.niem.gov (the original publication site) and GitHub (<https://github.com/NIEM/NIEM-Releases>).

[Definition: publication area]

An Internet location that stores core supplements, domain updates, code list updates, and other artifacts not produced during a NIEM release cycle, but are available for download and use with IEPDs. The NIEM publication area is <http://publication.niem.gov/niem/>.

[Definition: issue tracking area]

A tool or network location that supports registration, modification, and collaborative discussion of issues pertaining to NIEM schema documents, concepts, data components, and other artifacts. The issue tracking area is <https://github.com/NIEM/NIEM-Releases/issues>.

[Definition: collaboration area]

A tool or network location that supports cooperative editing of artifacts, as well as discussion by domains and other NIEM working groups. The issue tracker on GitHub is being used for this purpose.

3.3. Activities

There are five major activities involved in managing versions of NIEM. Domain representatives, NBAC, and NTAC engage in these processes as they improve and extend NIEM.

[Definition: domain update process]

A governance change process that results in a domain update. A domain may publish updates to its schema documents to the publication area. A domain update proposes updates to future NIEM release.

[Definition: core supplement process]

A governance process that results in a core supplement. NBAC may publish supplements to a NIEM core to the publication area. A core supplement is always additive or supplemental in nature; it CANNOT replace or modify a NIEM Core.

[Definition: domain reconciliation]

A governance process through which NBAC resolves conflicts between domain updates. This process results in a reconciled, coherent schema document set that is published as a minor release.

[Definition: cross-domain harmonization]

A process through which NBAC initiates tiger teams and working groups to resolve inconsistencies, overlaps, and other semantic issues between the domains, and between the domains and core. The results of these efforts will go into the issue tracking area, to be included in the next appropriate major or minor release.

[Definition: core synchronization]

The results of cross-domain harmonization are merged into a new release of the NIEM Core namespace, with domains synchronized to the new NIEM Core. Together, these form a major release.

3.4. A Walkthrough of Versioning

The NIEM versioning process contains several activities, performed by various parties. This section walks through these activities, and highlights the inputs and outputs of each. Each activity and artifact is discussed in more detail in later sections.

The following is a brief high-level walkthrough of versioning.

1. NIEM releases and domain updates are available for use in IEPDs and other use.
2. Through use and analysis of NIEM releases and published content, domain users will identify problems and new requirements for domain and core content. Users include developers of IEPDs, as well as implementers and

users of exchanges.

3. These problems and requirements will be entered and tracked as issues in the issue tracking area.
4. NIEM domains use issues as the basis for incremental improvements, extensions, and proposed changes to NIEM releases. This process is called the domain update process.
5. Domains publish their domain updates in two ways:
 - Issues are updated and new issues are entered into the issue tracking area.
 - Domain schema documents are updated in the publication area.
6. Periodically, the NBAC reconciles domain updates into a new minor release. The goals of this process are:
 - to resolve the conflicts and inconsistencies that are created by new and modified domain content
 - to produce a coherent set of reference schema documents for the domains
 - to perform some incremental harmonization on that schema document set. This process is called domain reconciliation, and is scheduled annually. In this process, the NBAC examines issues, published updates, and releases, and reconciles changes to domains into a single coherent domain schema document set. The NBAC ensures that all domains reference each other in a consistent way.
7. The result of this reconciliation process is published as the next minor release. This release is coherent. This release has a version identifier. This release is called a NIEM minor release, and it contains no modification to the NIEM Core namespace. It may contain some additional core content (in a new namespace), but will not modify the NIEM Core namespace. This is published annually, as the result of domain reconciliation. A minor release has a two-part numeric version identifier that does not end in 0, such as 4.2. For those issues that are not resolved via domain reconciliation, issues in the issue tracking area are updated, created, and closed, as appropriate.
8. At any time, asynchronously with other processes, NBAC or NTAC may initiate a tiger team to conduct cross-domain harmonization. A tiger team is a small working group of subject matter experts (SMEs) formed to solve a specific set of problems. A tiger team exists only for a short time; it is

disbanded once it has served its purpose.

9. A tiger team that is formed by the NBAC or NTAC will solve some set of problems with the NIEM model. For example, it may resolve a set of major harmonization issues between the domains and core. Or, it may address a new requirement that requires thought and experience. Or, it may identify, within the domains, components with semantic overlap, and will pull them up into core for common definition and use. This process is called cross-domain harmonization.
10. The results of tiger teams will be made available in a partition of the publication area. This partition may hold schema documents that contain additions to and extensions of the NIEM Core namespace. Each schema document is published with a unique namespace, distinct from the NIEM Core namespace. The domains may use these schema documents in domain updates, and IEPDs may use these schema documents, as well. The contents of these partitions of the publication area will be considered during the domain reconciliation process (step 6, above), and a supporting schema document containing additional core content may be published with the periodic minor release.
11. Every third year, the results of cross-domain harmonization and domain reconciliation are brought together into a new NIEM major release (instead of a minor release). This process is called core synchronization. The NIEM Core namespace is updated with help from resolved issues in the issue tracking area, as well as the published updates from the domains and tiger teams. Domain definitions that were dependent on the previous NIEM Core are harmonized with the updated NIEM Core namespace.
12. Core synchronization results in a new NIEM major release, with an updated NIEM Core namespace. A major release has a two-part numeric version identifier that ends in 0, such as 4.0.

4. Activities

This section contains further discussion and details of the major activities of the NIEM version architecture.

4.1. Domain Reconciliation

Once a NIEM release is published, updates and requests for changes accumulate. Users post issues (in the issue tracking area) describing problems with and unmet requirements of NIEM schema documents.

In response to these issues, domains publish domain updates to their namespaces. They identify and define new content that they require. They define and update code tables that they require. These updates are published in the domain partitions of the publication area.

The NBAC reconciles these requests with the latest published release, in order to produce a coherent NIEM schema document set. This reconciliation will result in a schema document set that is easily used by IEPD developers.

The NBAC makes decisions on how to accommodate changes. In the publication area, domains publish updates prior to NBAC approval, ensuring autonomy of the domains. The NBAC determines how best to migrate these updates into the next minor release. The NBAC conducts domain reconciliation annually.

This reconciliation process will consist of the following stages:

1. Gather and prepare input to the reconciliation process.

This input will come from open issues in the issue tracking area, and from domain partitions of the publication area. The content of the publication area will be evaluated for impacts. Reports will be prepared that may include:

- Reports describing the impacts of proposed changes on other domain namespaces.
- Reports assembling the comments that other domains have made about proposed changes.
- Reports containing technical metrics that quantify the
 - consistency of harmonization on the input schema document set, and
 - impacts of proposed changes.
- NTAC recommendations on technical methods and techniques for handling domain requests.
- Conformance to the [NIEM NDR](#) and quality checks. Conformance and quality checks are performed prior to publication of schema documents to the publication area, but additional reports on quality and conformance may be made available to the NBAC for reconciliation.

2. Reconcile proposed domain updates.

Starting with the previous NIEM release, the NBAC will review and apply the changes that were requested by the domains. For each proposed domain update, approve the update, or modify it as required to maintain high quality across all NIEM schema documents.

When a change is applied, make modifications in other domains, as required, to keep the schema document set coherent. Where domain updates introduce integrity issues, modify the schema document set to eliminate those problems. Propagate domain updates through impacted namespaces, as appropriate.

Note that there is no requirement or implication that any harmonization will be done by tools, or automatically in any way. The goal of this architecture is to ensure that the responsible party has the authority, schedule, goal, and ability to do what is needed to ensure that domain changes result in a coherent set of domain schema documents that can easily be used in IEPDs.

4.1.1. NBAC Decisions

Through the process of domain reconciliation, the NBAC will resolve many issues of cross-domain conflict. This may include changes to one domain that affect another domain, or overlaps of content between domains. Through the reconciliation process, the NBAC will make adjustments as needed to ensure that domain updates do not harm the quality of the NIEM release as a whole.

During its periodic domain reconciliation, the NBAC may conduct additional harmonization. Issues addressed may be low hanging fruit, such as particularly easy or notable problems. Or, the NBAC may try to resolve one or more aspects of harmonization across the NIEM model, such as vocabulary consistency. During reconciliation, it is likely that time will be short, and the level of effort will be limited. So, during reconciliation, extremely broad changes would be avoided, as would changes that required significant consideration by subject matter experts. Such issues would be handled by tiger teams and small groups in a separate process referred to as cross-domain harmonization.

The NBAC, as the designated body for harmonization, may make any decision that it deems prudent to integrate domain updates with the previous NIEM release. This may include the modification of dependent namespaces to accommodate domain changes, the creation of new elements for indirect (substitution) methods, and the modification of definitions and component names.

The NBAC may also choose to reject, in whole or in part, any domain update. The rejected domain content is still available (without change) in the publication area. However, it will not be part of domain reconciliation, and will not be included in future NIEM releases. The NBAC has the responsibility to act in the best interest of the entire NIEM community; it rejects domain content only to protect community interests, and it provides written justification when doing so.

4.1.2. Coherence

This document uses the term coherence to describe whether components of a schema document set refer to each other in a consistent manner:

[Definition: coherence]

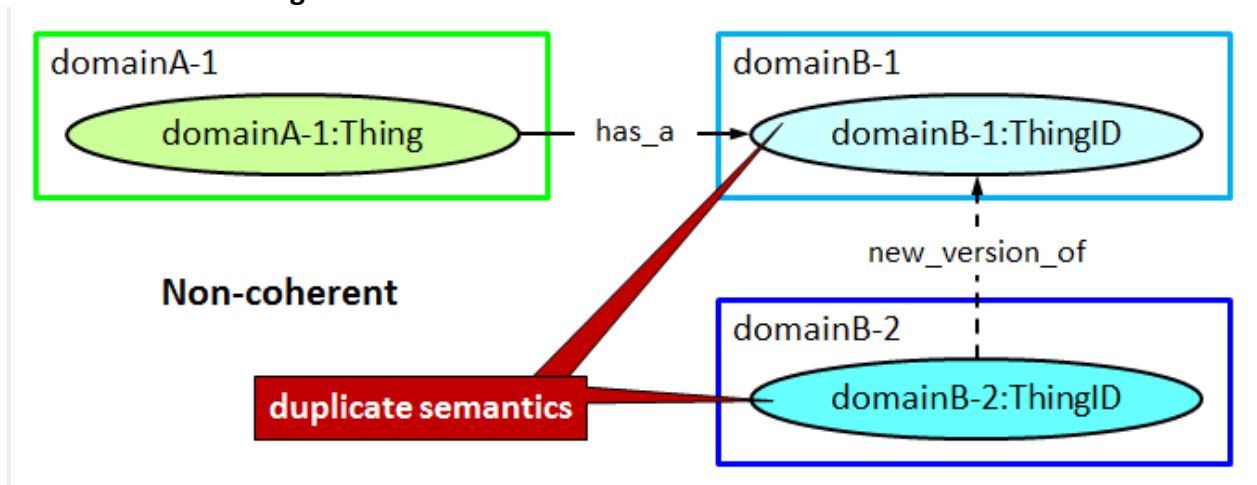
A schema document set is coherent when it has the following properties:

- It does not refer to a schema document outside the set, and
- It does not include two different versions of the same component in an incompatible way.

The coherence of a NIEM schema document set is best described by comparing a set that is not coherent with one that is. Imagine the following sequence of events, leading to the situation in Figure 4-1, *A schema document set that is not coherent*, below:

- Domain B publishes version 1 of its content, including a ThingID component.
- Domain A reuses that ThingID component within a Domain A component.
- Domain B publishes version 2 of its content, which contains a modified ThingID component.
- An IEPD developer wishes to reuse components from the latest versions of Domain A and Domain B. This results in the schema document set shown in Figure 4-1, *A schema document set that is not coherent*, below.

Figure 4-1: A schema document set that is not coherent



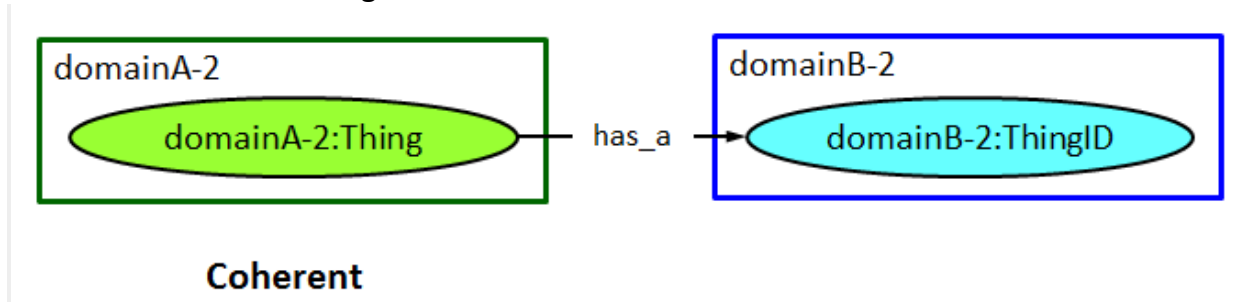
The schema document set in Figure 4-1, *A schema document set that is not coherent*, above, is not coherent, because it contains two versions of the same element: the new ThingID element in domainB-2, and the old ThingID element used in domainA-1. This is a duplication of semantics.

An IEPD with schema document sets that are not coherent can require significant work to make its schema documents work together. For example, the IEPD author might design an extension namespace to duplicate the changes to domainB-1:ThingID that distinguishes it from domainB-2:ThingID. These changes would likely cascade to related XML types, and similar changes would be required for each incompatibility between versions. Such an environment, in which many

domains are updating frequently, will ultimately present IEPD developers with substantial problems.

This problem will be resolved when Domain A publishes version 2 of its content, taking care to reuse components from version 2 of Domain B. Afterwards, the IEPD developer using the latest versions of Domains A and B will be working with the coherent schema document set shown in Figure 4-2, *A coherent schema document set*, below.

Figure 4-2: A coherent schema document set



Such a schema document set would not require custom extensions to fix problems related to incoherence. A later version of the version architecture specification will address specific rules for and the structure of version identifiers in the domain schema documents. For example, it may be desirable for version identifiers for bumped namespaces to be easily distinguished from version identifiers for domain-initiated changes.

4.2. Cross-Domain Harmonization

Larger and deeper issues of semantic consistency and modeling will be addressed by tiger teams and small groups of subject matter experts, as determined by the NBAC. The tasks to be undertaken by these cross-domain harmonization teams typically do not require participation from the entire NBAC. The use of such teams will enable parallel harmonization efforts. This will result in greater scalability, as the number of domains increase.

The products of cross-domain harmonization will include:

- Recommendations for domain updates in future minor releases, and
- Additional NIEM Core content provided via extensions and augmentations in new schema documents within the publication area.

The additions to NIEM Core that appear within the publication area may be immediately used by IEPDs, and domains may base published updates on those schema documents.

4.3. Core Synchronization

Periodically, these products will be synchronized with core content, to produce a new major release. A major release will contain a new NIEM Core namespace. A major release is the opportunity to change the existing core content by removing or altering core components. At the time of a major release, other major changes to the NIEM release schema document set may be considered by the NBAC, such as changes to structures and appinfo, which are schema documents that are part of the NIEM infrastructure. For more information on structures and appinfo, see the **[NIEM NDR]** and other documents on the NIEM Web site at <https://reference.niem.gov/niem/specification/>, also accessible via GitHub at <http://niem.github.io/reference/specifications/>.

NIEM Core synchronization will occur every third year. However, a NIEM Core synchronization may be held back for a longer period if changes to be released do not meet certain thresholds in terms of volume or severity. The NBAC must balance the need for regular scheduled releases with the set of components that must be harmonized across domains for inclusion into the core.

In between updates to NIEM Core, the NBAC may add core content via additional components in additional schema documents in a minor release of NIEM, without updating the NIEM Core schema document. These would be additive changes only, and would not modify the original NIEM core schema document in any way.

5. NIEM Releases

The NBAC domain reconciliation activity results in a NIEM minor release.

5.1. Properties of a NIEM Release

A NIEM release has the following properties:

- The release is coherent; that is, it contains only a single version of each domain namespace.
- The release integrates the latest input from the domains, with accommodations made by the NBAC to make them work together.
- The release contains new versions of only those namespaces that require update. Schema documents that require no update will be part of the release without modification.
- If required, a namespace will be updated to accommodate changes in other namespaces. This is determined by the NBAC during the reconciliation process.

5.2. Status of a NIEM Release

There are two types of harmonization/reconciliation:

1. between the domains (domain reconciliation), and
2. with NIEM Core and the domains (core synchronization)

This provides an opportunity to distinguish between the subsequent releases. Reconciliation of domains results in a minor release, while reconciliation of NIEM Core results in a major release.

5.3. Version Identifiers for NIEM Releases

Domain-synchronized NIEM releases are given minor release version numbers. NIEM Core-synchronized NIEM releases are given major release version numbers. Direct releases due to domain updates are given micro release numbers.

When core synchronization occurs, the results are published as a NIEM major release. These releases will be given identifiers such as 2.0, 3.0, and 4.0. These releases will yield a NIEM core namespace that has the same version identifier as the NIEM release.

When domain reconciliation occurs, without update of the NIEM core namespace, the results are published as a NIEM minor release. These releases have identifiers such as 2.1, 2.2, 2.3, 3.1, 3.2, and 3.3. When a domain update process occurs without reconciliation, the domain may request from NBAC a new release that incorporates that update, and the results are published as a domain update. These releases have identifiers such as 2.1.1, 2.1.2, 2.2.1, and 2.2.2.

For NIEM release identifiers, major releases have two integers (e.g., 2.0). The major integer (2), and the minor integer (.0). The minor release following 2.0 is 2.1. A domain update following 2.1 is 2.1.1, where the third integer indicates the domain update number. There is no version 2.1.0.

For NIEM release identifiers, the value following 2.9 is 2.10. Each number (major, minor, or domain update) is a field storing an integer, not a single digit.

The version identifiers for released domain schema documents (in the release area) and published schema documents (in the publication area) are not described by this HLVA. Constraints on these identifiers have not yet been determined, and will be specified fully in future version of the version architecture specification.

6. Data Stores

The version architecture defines several areas in which artifacts of NIEM are stored. Each may be implemented using new or existing tools. The goal of this document is to describe the areas,

what information they contain, and what functionality is needed to support activities that use these areas.

The specific software and tools that make up these data stores are not specified by this document.

6.1. The Release Area

All releases of NIEM are published to the release area.

All schema documents published to the release area are persistent, and are available for immediate use by IEPD developers and other users.

A single schema document may be a part of multiple releases. For example, a single release of the NIEM Core namespace will be a part of one or more minor releases. NIEM releases are now hosted on two sites: <https://release.niem.gov/> (the original publication site) and GitHub (<https://github.com/NIEM/NIEM-Releases/>).

6.2. The Publication Area

The publication area is a well-known location for domains and other working groups to publish schema documents and other artifacts for NIEM. This has two major purposes:

1. It allows published content to be immediately used.
2. It provides a concrete method for changes to be expressed for domain reconciliation and core synchronization.

All content of the publication area is persistent. That is, it never goes away once it is published. Also, it is never modified. Revised versions of published content may also be published, however, no artifact is removed or revised. This ensures that all content of the publication area may be freely used by any development effort, without fear of it changing or disappearing. All content of the publication area is available for use by IEPD developers, as well as developers of other schema documents and applications.

6.2.1. Partitions of the Publication Area

Each domain and working group controls a partition of the publication area. This partition is the publication site for products of the working group. The domain may place any number of schema documents into its publication area.

6.2.2. Use of the Publication Area

Schema documents within the publication area may provide additional content, or describe other changes to a NIEM release. Here is how the publication area is used:

- After a NIEM release is published, it becomes available for use by IEPDs. Through use and analysis of that NIEM release, domain users will identify problems and new requirements for domain content and core content. These will be conveyed via the issue tracking area.
- The domain, under its own governance, determines what changes it believes should be made to the latest NIEM distribution. It expresses these changes by publishing schema documents and supporting metadata in its publication area partition. Requests by the domain for changes to other domains and core will be made through the issue tracking area.
- Schema documents published to the publication area are immediately available for use by IEPDs. Both domain IEPDs and cross-domain IEPDs may use any schema documents in the publication area, as they require.
- The schema documents, supporting metadata, and issues (in the issue tracking area) describe changes the domain makes to its domain and requests for changes to core content. The requested changes are considered by the NBAC, and are reconciled with the last release, and with changes requested by other domains.
- The result of the NBAC reconciliation is published as the next NIEM release.

Schemas in the domain partition of the publication area may reuse, extend, and augment components within a NIEM release.

The schema documents in NIEM 2.0 contained schema documents that superseded the schema documents in NIEM 1.0. Schema documents in the domain publication areas are not required to supersede older schema documents in this manner. Instead, schema documents in the publication area may extend schema documents in previous NIEM distributions. This may make domain updates easier to use by IEPDs. For example, suppose the Justice domain wishes to add an element `ArrestSubject` to the type `ArrestType` that occurred in JXDM-4.0. It may publish a schema document in its publication area that augments or extends the JXDM 4.0 `ArrestType`. This extension may then add `ArrestSubject` to the `ArrestType`. A substitutable element may be made to provide for the new `ArrestType` to be used in IEPDs.

Extensions made in the manner above may be easier for IEPDs to use than replacement components. When a domain publishes an update as an extension of the latest release, that update may be easier for an IEPD to use than if it was published as a replacement of the latest release.

The specific method used in a domain update is left for the domain to determine. Some domains may choose to replace components, while others may extend components. It is

expected that best practices will be determined over time. Best practices may vary from domain to domain, as some domains may prefer to use common extensions, while others may prefer that each IEPD do its own integration work.

6.2.3. Namespaces for Content of the Publication Area

Each schema document that is posted to the publication area will have a namespace that meets several criteria.

- It is unique to the posted schema document. There will not be two schema documents posted to the publication area that have the same namespace URI.
- It clearly identifies that it is a namespace from the publication area.
- It clearly identifies the domain or working group that is responsible for it.
- It clearly identifies a version of the schema document target namespace. An example is `http://publication.niem.gov/niem/domains/cyfs/2.1/1/`, where the namespace is similar to that in a NIEM release, but with `publication` instead of `release`, and an additional number for the update version.

6.2.4. Additional Requirements of the Publication Area

Each schema document in the publication area has a target namespace that is distinct from that of all other schema documents. There will not be two different schema documents in the publication area with the same namespace, nor will there be a schema document in the publication area with the same namespace as a schema document in a NIEM release.

Each schema document in the publication area is persistent. Like the schema documents in NIEM releases, once published, the schema documents will not go away. These schema documents will remain available for use by IEPDs.

Schema documents in the publication area will be accompanied by change logs, which give comprehensible descriptions of the changes that were made to schema documents. This is discussed in Section 7.2, *Change Log*, below.

6.3. The Issue Tracking Area

NIEM now leverages GitHub to track ideas, enhancements, tasks, or bugs. This document uses the more general terminology issue tracking area rather than referring directly to GitHub.

The purpose of the issue tracking area is to capture and track requirements and problems from end users, developers, domains, and other people and bodies associated with NIEM. Issues may be entered by a helpdesk, in order to ensure that duplicates are not entered, and that issues are specified fully and clearly, so that additional work is not created downstream.

The issue tracker for release content issues is available at <https://github.com/NIEM/NIEM-Releases/issues/>. Note that issues pertaining to the Naming and Design Rules or other NIEM specifications should be submitted to the issue trackers within their own specific repositories.

7. Additional Artifacts

7.1. Change Log

[Definition: change log]

An artifact that describes the changes applied to a release since its previous version.

The publication area will contain domain updates. These updates are provided as XML Schema documents, and will be accompanied by a change log.

A change log is able to express a single big change that is composed of a set of small changes. For example, the renaming of an element may include changes to many types that used that element.

Without a change log, changes made between versions of a schema document would not include an expression of what the changes to the schema document might mean. Although the revised schema document may be fully formed and well defined, it may be unclear what impact the change might have on other schema documents.

For example, upon renaming an element (from an old name to a new name), the change is represented in XML Schema by the original element (with the old name) not appearing in the revised schema document, and a new element (with the new name) appearing in the revised schema document.

A type in another domain may be impacted by this change. If the type used the original element with the old name, it would be unclear what changes should be made to the type to accommodate the renaming.

In order to describe changes to schema documents with enough detail to enable reconciliation, schema document definitions are accompanied by metadata that describe changes. This metadata will include a list of changes, without restating the details that occur in the schema documents that are in the publication area.

The mechanics and format of the change log are to be determined. The format will be in XML, and may include XPath expressions for relating change log entries to updated schema

documents. The change log XML file may describe the following about each change made by a domain:

- A description of the change, which says what was done and, perhaps, why it was done.
- A list of the components that were added as a result of the change. It may include the location of each new component, via XPath locations. It may also describe the location (in the sequence of the original component) where an element may be inserted into a type.
- A list of the components that were deleted as a result of the change. (Note that no change actually deletes anything as all schema documents are persistent, but deleted components will not exist in future schema documents)
- A list of modifications, each modification giving the location of the original definition and the location of the new definition.

Users will be able to audit change logs against schema document updates to look for

- changes and updates that are inadequately described by the change log, and
- changes in the change log that is not correlated by updated content. Such a change log may be audited when uploaded, before publication of the schema documents and change log to the publication area.

Descriptions of the changes made are provided to the NBAC during reconciliation, and would help them to determine the contents of the next NIEM release.

7.2. Namespace URI Directive

Through the NBAC's domain reconciliation process, it may be determined that some namespaces must be adjusted to accommodate changes to other namespaces. That is, although the domain has no new updates for its namespace, the NBAC must change the domain namespace to accommodate changes in other domains. In this situation, we say that the domain namespace was bumped.

[Definition: bump]

A process whereby a domain schema document requires a new target namespace URI as a result of one or more changes made to another NIEM domain schema document that impact the content of the schema document to be bumped.

The version architecture does not allow the application of changes directly to a published namespace. A changed schema document must be given a new target namespace URI. So, when a domain namespace is bumped, it must be given a new namespace URI.

[Definition: namespace URI directive]

An artifact that specifies the new target namespace URI for a new schema document resulting from the need to bump it.

The exact mechanics of how a namespace URI directive is published is to be determined. One option is for it to be an XML file that is published in the publication area, within the domain's partition.

There are several goals for the namespace URI directive:

- to provide clear direction to those building a NIEM release regarding how to construct namespaces for domain schema documents
- to enable domains to maintain control of their namespace URIs, and the progression of those URIs from version to version. It is expected that a domain may wish to exert strong control over the version identifiers for its schema documents.

The namespace URI directive allows the domain to express the structure of its target namespace URIs, and the sequence of version identifiers. The domain will be able to specify how namespace URIs are incremented when the namespace is updated or bumped.

8. General Release Timelines

The NIEM version architecture is designed to be agile, accommodate regular change, and respond efficiently to stakeholder requirements. To do so, NIEM maintains the ability to execute predictable, repeatable, and consistent release cycles, while also allowing its domains to publish updates independently (on their own timelines). At a high level, this section describes the timelines, coordination, and options for major and minor release cycles.

[Definition: release cycle]

A period of time during which appropriate groups of NIEM governance and domain representatives engage in the activities that result in a major or minor NIEM release.

NIEM will generally publish an annual release in the same quarter of each year. The normal target is the third quarter of the Government fiscal year (subject to change as determined by NIEM governance and the NMO). Within each three-year period, the NIEM program will execute two consecutive minor release cycles, followed by a major release cycle.

A release date will be scheduled (internally established by the NMO) for the beginning of the target quarter. The release date may shift later in the quarter if necessary. This adds a degree of flexibility and safety to the schedule to accommodate additional pre-release phases (e.g., beta2

or beta3) or other unforeseen events, while still maintaining consistency and predictability. The NMO will communicate anticipated scheduling details to the NIEM community for planning.

If required to meet schedule, a release cycle may be time-boxed or content-scoped (or both).

For each release cycle, the NMO will assign a release coordinator selected from the major stakeholders. The release coordinator will monitor and report progress to NMO, NTAC, NBAC, and domains, resolve issues of schedule and scope, and coordinate participating entities (NIEM governance, domains, release manager, etc.).

Ordinarily NIEM will not issue a minor release for a previous major version series (such as version 2.0). However, a request by a domain (or other entity) to issue a minor release for such (e.g., a request for a Version 2.3) will be jointly considered by NMO, NBAC, and NTAC. If approved, it will be executed between annual release cycles (since the scope of change is expected to be small).

8.1. Release Cycle Stages

A typical NIEM release cycle is comprised of three basic stages of development, each of which may require several iterations. These are defined as follows:

[Definition: alpha]

The initial stage of a release cycle. Primary activities focus on integrating architectural improvements (if any) as well as domain and core content changes previously prepared as domain and core supplements since the last release. Other activities include reconciliation, core synchronization (when appropriate), appropriate types of harmonization, refactoring/refinement of content as needed, application of resolved issues, and iterative NIEM governance reviews and feedback. The most significant types, volume, and scale of changes occur during alpha stage.

[Definition: beta]

The midterm stage of a release cycle. At the beginning of this stage the developing release has stabilized to the degree that remaining changes are minor — major adjustments for architecture have been completed; no significant new content may be integrated; release schemas are in a state that will allow user beta testing (i.e., the release is expected to be NIEM-conformant). Changes are still allowed; however, these are much smaller in magnitude and complexity and are generally only corrections and adjustments. Examples of beta stage changes include adjustments to data component structure; movement of properties; renaming and redefining; addition or refinement of keyword, usage, and example metadata (associated with definitions); addition of a very small number new components (essentially inadvertent omissions or additions needed to adjust structure), and occasionally very minor corrections to

architecture. If beta testing, reviews, and feedback discover major flaws that require significant changes, then the release may revert back to the alpha stage.

[Definition: release candidate]

The final stage of a release cycle. A release candidate should be identical to, and therefore is a potential final production (i.e., operational) release. Only extremely minor corrections to a release candidate are allowed. These include minor corrections to character strings, names, definitions, namespaces, etc. At this stage absolutely NO architectural or significant content changes are allowed. Any such changes will require regenerating the release as a beta product. Activities during this stage are essentially only testing, review, feedback, and possibly very minor refinements. As this is the final review, diligence must ensure quality and conformance before publication. If testing, reviews, and feedback discover flaws that require significant changes, then the release may revert back to the beta, or even alpha stage. Very minor refinements to a release candidate will require regenerating it as a new release candidate.

In the past, NIEM has often referred to a so-called pre-alpha stage. In fact, this is not a formal stage of a NIEM release cycle.

[Definition: pre-alpha]

Any time period during which the NIEM program is NOT executing a release cycle, i.e., the time period after the last release cycle ends to before the next begins. Pre-alpha activities may include (but are not limited to): production of domain updates and core supplements; identification and resolution of issues, identification of new requirements, refactorization of NIEM objects, and essentially any other activity that can be performed in a preparatory manner before a release cycle begins, and recorded into the issue tracking area and subsequently applied in the next appropriate release cycle.

Pre-alpha activities significantly reduce the surge activity often perceived or occurring with a release cycle.

8.2. Release Cycle Exceptions and Options

Based on history, two minor releases and one major release in a three-year period (each release cycle executed annually) will generally accommodate the expected demand for future model changes. So, by default the appropriate release will always be planned each year without the need for prior approvals.

There is no desire to force unnecessary releases. Therefore, as circumstances demand, the NTAC, NBAC, domains, and the NMO may jointly evaluate conditions and elect to adjust the schedule as needed. Special options include:

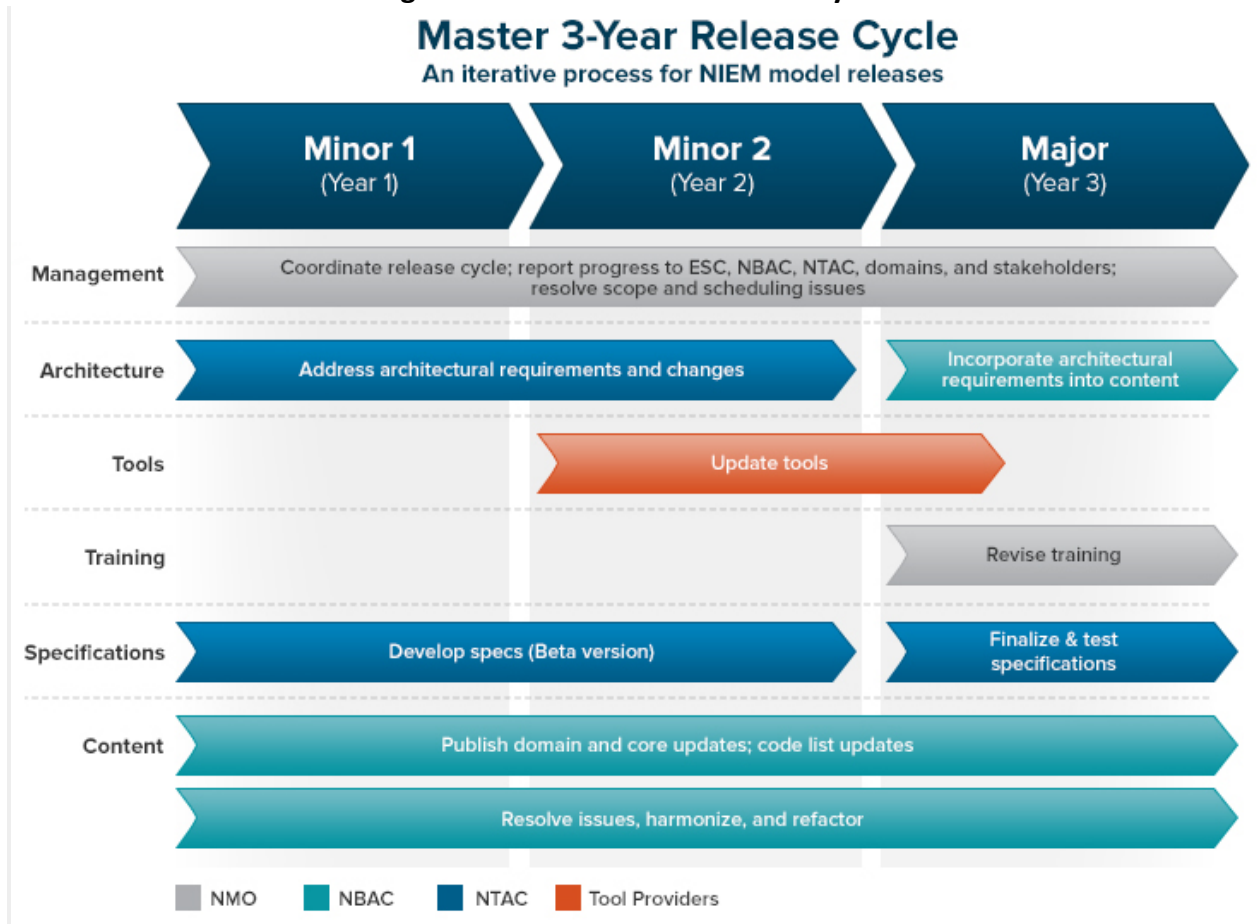
- Cancel a minor release (e.g., due to insufficient need for domain and code table changes).
- Postpone a major release for a year (e.g., due to insufficient architectural and core changes). In this case, the expected major release may be replaced by a minor release to accommodate regular domain and code table changes.
- Advance a major release (e.g., due to high priority architectural or core changes). In this case, the regular minor release would be cancelled.

8.3. Outline of the Three-Year Cycle

Figure 8-1, *The NIEM Three Year Cycle.*, below, illustrates a typical NIEM three-year period. The following notes explain the activity tracks within the diagram:

- Architecture and specifications: Two years are available to address and prepare for architectural requirements and develop associated specification changes prior to the next major release during which such changes are applied.
- Content: A normal NIEM release cycle begins with the alpha stage. Per this HLVA, whenever NIEM is not executing a major or minor release cycle, it is in a pre-alpha stage. Pre-alpha includes adding and resolving content issues ; preparing domain updates; on-boarding new domains; addressing new requirements; harmonizing content and queuing solutions for the next release; etc.
- Tools: After a year, new or changes to existing specifications will enter a more stable beta stage and can be published for the community review. Upgrades to reference tools and other capabilities can begin development and finish in time to assist with the next major release.
- Training: Training can begin revision to keep pace with updates to architecture and content.
- Face-to-face meetings: Though not shown in this chart, the annual joint NBAC/NTAC face-to-face meeting can coincide with initial planning (or at least as a key synchronization point) for the release cycle.

Figure 8-1: The NIEM Three Year Cycle.



Although face-to-face governance meetings are extremely valuable for release planning, coordination, harmonization, and issue resolution, they are not absolutely necessary. In the event that face-to-face meetings cannot be held, teleconferences, collaboration tools, and other means will be sufficient to execute a release.

9. Summary

Under the NIEM version architecture, domains are free to publish updates on their own timelines, without being required to wait for NIEM governance or other domains to act. The architecture ensures that all such updates are available for use in IEPDs or other uses of NIEM schema documents. The architecture ensures that NIEM releases will be available to developers at regularly scheduled intervals, and that those releases will be easy to use, and free of structural inconsistency.

The version architecture also ensures that domain changes are available in a concrete form that may be incorporated into the next NIEM release. It also ensures that NIEM releases are scheduled such that they may incorporate recent domain changes. NIEM governance has

authority to make required modifications in order to ensure consistency between the domains and core.

Appendix A. Acronyms and Abbreviations

Acronym / Abbreviation	Literal or Definition
ESC	Executive Steering Council
HLVA	High Level Version Architecture
IEP	Information Exchange Package
IEPD	Information Exchange Package Documentation
NDR	Naming and Design Rules
NIEM	National Information Exchange Model
NBAC	NIEM Business Architecture Committee
NTAC	NIEM Technical Architecture Committee
NMO	NIEM Management Office
SSGT	Schema Subset Generation Tool
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSD	XML Schema Definition

Appendix B. References

- [NIEM Conformance]:** *NIEM Conformance, Version 3.0*, NIEM Technical Architecture Committee (NTAC), 15 August 2014. Available from <http://reference.niem.gov/niem/specification/conformance/3.0/>.
- [NIEM Conformance Target Attribute Specification]:** *NIEM Conformance Targets Attribute Specification, Version 3.0*, NIEM Technical Architecture Committee (NTAC), 31 July 2014. Available from <http://reference.niem.gov/niem/specification/conformance-targets-attribute/3.0/>.
- [NIEM Domain Update Specification]:** *NIEM Domain Update Specification, Version 1.0*, NIEM Technical Architecture Committee (NTAC), 5 November 2010. Available from <http://reference.niem.gov/niem/specification/domain-update/1.0/>.
- [NIEM High-Level Tool Architecture]:** *NIEM High-Level Tool Architecture, Version 1.1*, NIEM Technical Architecture Committee, 1 December 2008. Available from <http://reference.niem.gov/niem/specification/high-level-tool-architecture/1.1/>.
- [NIEM High-Level Version Architecture v1.0]:** *NIEM High Level Version Architecture (HLVA)*, Version 1.0, NIEM Technical Architecture Committee, 31 July 2008. Available from <http://reference.niem.gov/niem/specification/high-level-version-architecture/1.0/>.
- [NIEM High-Level Version Architecture v3.0]:** *NIEM High Level Version Architecture (HLVA)*, Version 3.0, NIEM Technical Architecture Committee, 27 April 2015. Available from <http://reference.niem.gov/niem/specification/high-level-version-architecture/3.0/>.
- [NIEM Implementation Guide]:** NIEM Implementation Guide, NIEM Program Management Office. Available from <https://www.niem.gov/aboutniem/grant-funding/Pages/implementation-guide.aspx>.
- [NIEM MPD Specification 3.0.1]:** *NIEM Model Package Description (MPD) Specification, Version 3.0.1*, NIEM Technical Architecture Committee (NTAC), 27 April 2015. Available from <http://reference.niem.gov/niem/specification/model-package-description/3.0.1/>.
- [NIEM MPD Toolkit]:** NIEM Model Package Description Toolkit, Version 3.0, NIEM Technical Architecture Committee (NTAC), 15 August 2014. Available from <http://reference.niem.gov/niem/resource/mpd/3.0/>.

[NIEM NDR]: *NIEM Naming and Design Rules (NDR)*, Version 3.0, NIEM Technical Architecture Committee (NTAC), 31 July 2014. Available from <http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/>.

[NIEM Quality Assurance]: *NIEM Quality Assurance Strategy and Plan*, Version 1.0, NIEM Technical Architecture Committee (NTAC), 20 May 2008. Available from <http://reference.niem.gov/niem/guidance/quality-assurance-strategy-and-plan/1.0/>.

[PKZIP]: *APPNOTE.TXT - .ZIP File Format Specification*, Version: 6.3.2, Revised: 28 September 2007, Copyright (c) 1989 - 2007 PKWare Inc. Available from <http://www.pkware.com/documents/casestudies/APPNOTE.TXT>.

[RFC 2119 Key Words]: 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>.

[RFC 3986 URI]: Berners-Lee, T., et al., *Uniform Resource Identifier (URI): Generic Syntax*, Request for Comment 3986, Network Working Group, January 2005. Available from <http://tools.ietf.org/html/rfc3986>.

[W3-XML]: *Extensible Markup Language (XML)*, Version 1.0, Fifth Edition, W3C Recommendation 26 November 2008. Available from <http://www.w3.org/TR/2008/REC-xml-20081126/>.

[W3-XML-InfoSet]: *XML Information Set*, Second Edition, W3C Recommendation 4 February 2004. Available from <http://www.w3.org/TR/2004/REC-xml-infoset-20040204/>.

[W3-XML-Namespaces]: *Namespaces in XML*, Second Edition, World Wide Web Consortium 16 August 2006. Available from <http://www.w3.org/TR/2006/REC-xml-names-20060816/>.

[W3C XML Schema Datatypes]: *XML Schema Part 2: Datatypes*, Second Edition, W3C Recommendation 28 October 2004. Available from <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>.

[W3C XML Schema Structures]: *XML Schema Part 1: Structures*, Second Edition, W3C Recommendation 28 October 2004. Available from <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>.

Appendix C. Index of Definitions

- [alpha](#): Section 8.1, *Release Cycle Stages*
- [beta](#): Section 8.1, *Release Cycle Stages*
- [bump](#): Section 7.3, *Namespace URI Directive*
- [change log](#): Section 7.2, *Change Log*
- [coherence](#): Section 4.1.2, *Coherence*
- [collaboration area](#): Section 3.2, *Data Stores*
- [core supplement](#): Section 3.1, *Classes of Information*
- [core supplement process](#): Section 3.3, *Activities*
- [core synchronization](#): Section 3.3, *Activities*
- [cross-domain harmonization](#): Section 3.3, *Activities*
- [domain reconciliation](#): Section 3.3, *Activities*
- [domain update](#): Section 3.1, *Classes of Information*
- [domain update process](#): Section 3.3, *Activities*
- [harmonization](#): Section 3, *Overview*
- [issue tracking area](#): Section 3.2, *Data Stores*
- [major release](#): Section 3.1, *Classes of Information*
- [minor release](#): Section 3.1, *Classes of Information*
- [namespace URI directive](#): Section 7.3, *Namespace URI Directive*
- [NIEM core](#): Section 1.2, *Background*
- [NIEM domain](#): Section 1.2, *Background*
- [NIEM governance](#): Section 1.2, *Background*
- [NIEM issue](#): Section 3.1, *Classes of Information*
- [NIEM release](#): Section 3.1, *Classes of Information*
- [pre-alpha](#): Section 8.1, *Release Cycle Stages*
- [publication area](#): Section 3.2, *Data Stores*
- [published update](#): Section 3.1, *Classes of Information*
- [release area](#): Section 3.2, *Data Stores*
- [release candidate](#): Section 8.1, *Release Cycle Stages*
- [release cycle](#): Section 8, *General Release Timelines*

Appendix D. Revision History

Significant changes in version 3.0 (27 August 2015):

1. This document is now in HTML format (and text) with highlighted definitions and hyperlinks.
2. Added new Section 8, *General Release Timelines*, above, describing the NIEM Release Optimization Strategy approved by the Executive Steering Council (ESC) in June 2014.
3. Revised both Figure 4-1, *A schema document set that is not coherent*, above, and Figure 4-2, *A coherent schema document set*, above, for clarity. Text referring to these figures was also (1) revised for clarity, and (2) better synchronized to the figures.
4. All former references to "core update" are now core supplement.
5. Added to Section 1.1, *Scope*, above, that the HLVA does not address IEPD versioning. IEPDs are managed and versioned independently of the Core and domains. In the near term, the PMO will publish guidance for IEPD change control and versioning (through the NIEM Technical Architecture Committee).