

Tool Strategy Requirements Document

(v116)

Exported on Mar 19, 2020

Table of Contents

1 Overview 4

2 Purpose 5

3 Scope 6

4 Assumptions 7

5 Boundaries 8

6 Constraints 9

7 Risks 10

8 Functional Requirements 11

8.1 Tool Infrastructure 11

8.2 IEPD Lifecycle Support 13

8.2.1 Scenario Planning 13

8.2.2 Analyze Requirements 14

8.2.3 Map and Model 15

8.2.4 Build and Validate 17

8.2.5 Assemble and Document 18

8.2.6 Publish and Implement 19

8.3 Browse and Search 19

8.4 Support Multiple Formats 20

8.5 Interfaces 23

8.6 Help System 23

8.7 Backlog 23

9 Non-Functional Requirements 25

10 Use Cases 27

10.1 UC1 27

10.1.1 Purpose 27

10.1.2 Actors 27

10.1.3 Pre conditions 27

10.1.4 Post conditions 27

10.1.5 Main Flow 27

10.1.6 Exceptions 27

11 Next Steps 28

12 References 29

13 Appendix 30

13.1 NBAC Annual Report - Tool Recommendations 30

13.2 Requirements Discussion (GTRI Site Visit) 30

14 Glossary 31

15 Acronyms 33

**DRAFT - FOR INTERNAL USE ONLY**

# Overview

In January 2019, the U.S. Department of Defense (DoD) became the official government sponsor of the National Information Exchange Model (NIEM). One of the high-priority projects identified during the transition period and subsequent meetings with stakeholders has been the development of a comprehensive Tool Strategy (TS) to efficiently facilitate widespread adoption and use of NIEM.

NIEM requires a framework that supports tool interoperability through standard interfaces and well-defined artifacts. This will enable the NIEM Management Office (NMO) to leverage its limited resources to more rapidly expand automated support for development of NIEM exchanges.

*(To be continued)*

# Purpose

The goal of the NIEM TS is to provide an end-to-end tool capability that facilitates and automates all phases of the NIEM IEPD Lifecycle to include:

* tracing information exchange requirements from scenario planning architecture artifacts to mapping organic data stores to the NIEM reference schema
* developing new NIEM-conformant extension schemas
* supporting application program interface (API) development
* posting IEPDs for discovery and facilitating reuse

Due to budget constraints, the NIEM TS balances community needs against tool complexity and feasibility with a phased development approach that will result in a maximum return on investment to:

* Maximize information interoperability and reuse.
* Minimize the cost of using NIEM (particularly entry costs).
* Reduce time to develop IEPDs.
* Minimize the cost of increasing automated support for NIEM.
* Leverage more existing and future open-source tool resources.
* Encourage more tool developers to build NIEM tools and capabilities.
* Maximize tool integration and inter-operation opportunities.
* Standardize NIEM processes so they are measurable and repeatable.
  + Maximize consistency and quality of release products, IEPDs, and associated artifacts.
  + Minimize NIEM training requirements and technical expertise needed to use NIEM.
  + Maximize domain, developer, and user self-service.

# Scope

*(Add intro here on overall scope)*

The Information Exchange Package Documentation Life Cycle (IEPDLC) is the primary process for development of the artifacts that define an information exchange specification. The IEPDLC provides a guide to understanding how IEPDs are built from NIEM and published. It is not intended to be prescriptive. IEPD builders may enter the life cycle at any particular step, as well as adjust the scope of the life cycle to support the level of effort required for their individual IEPD development. This perspective identifies tool interfaces required in building and implementing NIEM IEPDs.

From a user perspective, the NIEM TS addresses the functional tool needs for the following user classes:

* Stakeholders or stakeholder communities with an interest in NIEM.
* Practitioners including IEPD developers and implementers.
  + NMO, technical assistance staff, content management staff, and development staff.
  + Governance bodies including NTAC, NBAC, domain stewards and associated committees, and tiger teams.
  + Tool developers including commercial and government developers, contract developers affiliated with the NMO, and development staff.

# Assumptions

* Development and deployment of new NIEM Tools will not delay the NIEM Release schedule.
* The following tools will still be maintained during new Tool development:
  + [**Schema Subset Generation Tool - SSGT**](https://tools.niem.gov/niemtools/ssgt/index.iepd)**:** enables users to search and explore the content of the NIEM model. Additionally, users have the option of building XML Schema subsets of a NIEM release for use in NIEM XML exchanges. Based on the list of components selected for the subset, the tool will calculate dependencies and generate a valid set of schemas that are a subset of a release for download as a zip file.
  + [**Conformance Testing Assistant - ConTesA**](https://tools.niem.gov/contesa/)**:** enables users to test NIEM XML schema's against the automated rules from the NIEM Naming and Design Rules (NDR).
  + [**Movement**](https://beta.movement.niem.gov/)**:** enables users to search and explore the content of the NIEM model. It provides a user friendly interface, smarter search results, and a streamlined way to build JavaScript Object Notation (JSON) Schema in support of exchanges.
  + [**Migration Assistance**](http://tools.niem.gov/niemtools/migration/index.iepd)**:** enables users to upgrade an XML Schema release subset to a newer version. This supports the process to update an existing exchange to use a more recent NIEM release.
* The development of new NIEM Tools (funded by government sponsors) will be open-source.
* Tools will be released on an independent schedule as completed.
* Tools will be modular and decoupled from the NIEM Schema.

# Boundaries

* The Tool Strategy captures requirements specifically related to new open-source tool development. All other current and future (e.g. IEPD Registry) tools are out of scope.

# Constraints

* Budget - additional funding has not been secured to develop the tools, therefore the tools will be developed within the existing budget.
* Resources
  + Development of Tools
    - Implementation
    - Hosting
    - Support
    - Timeline

# Risks

* NIEM Management Office (NMO) funding is uncertain in future fiscal cycles.
  + **Mitigation:**Tools will be developed in open-source modules and/or micro-services, so that they can be leveraged independent of one central software application.
* NIEM will eventually transition to a new government sponsor, possibly during a future Tool development phase.
  + **Mitigation:** Continuous documentation included with each MVP and/or Phase of project.

# Functional Requirements

The requirements were generated using the following language for consistency in communication.

**Format**

All requirements are identified by a unique format as described below:

* **NIEM-#**
  + **NIEM** - Project Name
  + **#**  - Unique Identifier

Following this format will allow these requirements to be rearranged as needed and are not specific to this document. In addition, the requirements can be further 'decomposed' by adding a sub-level number under each Level 1 requirement. It should be noted that the number assignment, order, and categorization of the requirements are not intended to convey any dependency and/or sequence of development.

**Definitions**

* **MUST** – This word, or the terms "**REQUIRED**" or "**SHALL**", mean that the definition is an absolute requirement of the specification.
* **MUST NOT** – This phrase, or the phrase "**SHALL NOT**", mean that the definition is an absolute prohibition of the specification.
* **SHOULD** – This word, or the adjective "**RECOMMENDED**", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
* **SHOULD NOT** – This phrase, or the phrase **"NOT RECOMMENDED**" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

## Tool Infrastructure

Developers and Project Managers will work to adopt an Agile software development methodology that leverages industry-standard best practices. In this stage, the Developers will research and identify the hardware/software needs for designing the core components of the tool requirements.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-103**](https://jira.di2e.net/browse/NIEM-103?src=confmacro) **- The tool shall include a default landing page for user access.  to do**

The initial increment of the tool shall be the underlying tool infrastructure with a simple landing page developed in the chosen front-end framework.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-104**](https://jira.di2e.net/browse/NIEM-104?src=confmacro) **- The tool shall be a containerized application.  to do**

The tool shall be containerized to allow for deployment to a publicly-hosted site, multiple privately-hosted secure sites, and user environments as needed.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-105**](https://jira.di2e.net/browse/NIEM-105?src=confmacro) **- The developer shall document a DevSecOps methodology for project management.  to do**

The developer shall follow a process to release tool features incrementally, collaborate with customers over functionality, develop with application security in mind from the start, and automate the application builds, testing, and deployment.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-106**](https://jira.di2e.net/browse/NIEM-106?src=confmacro) **- The developer shall create technical documentation that outlines how third-party developers can interact with software code base.  to do**

The tool shall be designed so that outside developers can contribute code to the overall project or contribute specific widgets like custom translation modules.  The developer shall design for this future capability at the outset.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-107**](https://jira.di2e.net/browse/NIEM-107?src=confmacro) **- The tool shall leverage automatic use of Local Storage within a client-side browser to persist data and state.  to do**

The tool shall be designed to leverage automatic use of Local Storage capabilities within a client-side browser to persist data and state so that work is not lost if the application is terminated and/or restarted.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-108**](https://jira.di2e.net/browse/NIEM-108?src=confmacro) **- The tool shall leverage industry-standard security measures for scanning any external files uploaded by an end user.  to do**

The tool shall scan for malicious code in uploaded files.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-109**](https://jira.di2e.net/browse/NIEM-109?src=confmacro) **- The tool shall incorporate session management to facilitate authentication and access control capabilities.  to do**

Two (2) main features would include:

* Auto-saving work (state saving) as the user fills out web forms
* Collecting audit information about operations the user performs (e.g. create, read, update and delete - CRUD operations)

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-110**](https://jira.di2e.net/browse/NIEM-110?src=confmacro) **- The tool shall incorporate a user account registration and management capability.  to do**

Users shall be able to create accounts in order to later store uploaded artifacts, entered data, and settings.  The NIEM Management Office shall be able to perform private administrative functions.

* The user account registration should require verification prior to activating a new user profile.
  + **Note:** The developer should define the verification method (e.g. email).
* The user account registration should be configurable, so that a user can toggle this functionality on/off if unnecessary for their local and/or offline deployment.

The following table describes a high-level overview of the proposed role-based access control (RBAC) capabilities:

|  |  |
| --- | --- |
| **Role Type** | **Permissions** |
| General User | * Account Registration * Ability to update their own account profile (e.g. email address) * Ability to create, read, update, and delete their own data * Ability to upload and delete their own artifacts (e.g. mapping spreadsheet, supporting documentation) * Ability to submit Contact form to request NIEM Management Office (NMO) assistance * Ability to import and export valid NIEM Schemas * Ability to browse and search NIEM components * Ability to add custom NIEM components (e.g. augmentations) * Additional features described within this document |
| Administrator | * *All permissions of a General User* * Ability to create, read, and update General User & Administrator Account profiles * Ability to review Contact form submissions * Ability to create, read, update, and delete custom NIEM components submitted by General Users |
| Super Administrator | * *All permissions of a Administrator* * Ability to review Audit logs of all General User & Administrator CRUD activity * Ability to review Audit logs of any developed API services * Ability to create, read, update, and delete all User Account profiles * Ability to create, read, update, and delete NIEM Core and Domain components |

Table 1: Overview of Basic Role Types and Permissions

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-111**](https://jira.di2e.net/browse/NIEM-111?src=confmacro) **- The tool shall support both NIEM release and published NIEM community content.  to do**

The tool shall provide the similar support for NIEM releases and for community-provided content that passes conformance tests.  This community content includes EIEMs, IEPDs, new domain submissions, domain updates, and other local models.  Features like searching, browsing, reusing components in subsets, and generating documentation should work for any kind of NIEM content, not just content from NIEM releases.  Note that this will require the option to allow community content published to the public NIEM IEPD Repository or built with this tool to be imported and stored.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-112**](https://jira.di2e.net/browse/NIEM-112?src=confmacro) **- The tool shall provide initial NIEM support beginning with the NIEM 3.0 release.  to do**

The NIEM 3.0 release came with several major changes.  It introduced the conformanceTargets attribute, which NIEM uses to indicate the applicable set of NDR conformance rules for NDR conformance validation.  NIEM 3.0 changed the way schemas were represented, including the introduction of augmentations and the removal of appinfo in component annotations.  And the NIEM NDR 3.0 was significantly restructured to provide actual Schematron**1** representations of rules where possible to allow for easier and more consistent conformance checking.  Because of these key differences between NIEM 3.0 schemas and prior schemas and the level of work it would take to support the older releases, initial tool support should begin with NIEM 3.0.  Support for NIEM 1.0, 2.0, and 2.1 can be provided later on based on priority and user demand.

## IEPD Lifecycle Support

The NIEM IEPD Lifecycle consists of six phases that walk users through designing, developing, publishing, and maintaining an IEPD.  The tool should be able to guide users through each phase of development, filling in existing gaps in IEPD Lifecycle support and also supporting artifacts generated outside the tool.



Figure 1: Phases of the IEPD Lifecycle

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-113**](https://jira.di2e.net/browse/NIEM-113?src=confmacro) **- The tool shall provide a guided path for IEPD development as well as direct access to key functionality.  to do**

The tool shall provided a guided walk-through for users building an IEPD, allowing them to start at the beginning and providing guidance or full support through the full IEPD Lifecycle.  Users shall not be limited to this assisted walk-through, however, but shall be allowed to drop in to the lifecycle as needed.  As some users may choose to build tool-supported artifacts in their own external tools, this tool shall also accommodate direct access to key features.

### Scenario Planning

A user reviews background information related to their information exchange, assesses resource impact, understands business context, and identifies information exchange business scenarios. While this is mainly done outside of NIEM, the tool should allow users to upload supporting documentation as well as enter information about an IEPD.  This will feed into the generation of the MPD catalog during the later *Assemble and Document* phase.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-114**](https://jira.di2e.net/browse/NIEM-114?src=confmacro) **- The tool shall include the ability to upload scenario planning artifacts.  to do**

The *Scenario Planning* lifecycle phase typically involves the creation of UML diagrams, documents, and other artifacts to capture things like information exchange use cases, business processes, exchange partners, and message flows.  Many existing tools already provide this functionality, which does not need to be duplicated.  This tool shall allow users to upload these artifacts so that they may be later bundled into the IEPD package in the *Assemble and Document* lifecycle phase.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-115**](https://jira.di2e.net/browse/NIEM-115?src=confmacro) **- The tool shall include the ability to enter basic IEPD metadata into a web form.  to do**

A MPD catalog is a required artifact in an IEPD.  This is a XML file that documents general exchange metadata (like IEPD name, description, and contact information) and also provides information about each artifact in the package (like what kind of artifact it is and its file path).  Information needed to fill in the general metadata portion of the MPD catalog should be discussed during the *Scenario Planning* phase and should be captured here via a form.  These values should be saved for the later generation of the MPD catalog in the *Assemble and Document* lifecycle phase.

### Analyze Requirements

The selected information exchange scenario is further elaborated to understand and document the business context and data requirements. Users should be able to upload supporting documentation and download/upload a standardized data mapping spreadsheet to enter data requirements.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-116**](https://jira.di2e.net/browse/NIEM-116?src=confmacro) **- The tool shall include the ability to upload data requirement artifacts.  to do**

Exchange data requirements and business rules may be provided in a variety of formats, not just via the NIEM mapping spreadsheet.  Formats include documents, custom spreadsheets, UML diagrams, and database schemas.  The tool shall allow users to upload any such artifact that documents data requirements in this phase.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-117**](https://jira.di2e.net/browse/NIEM-117?src=confmacro) **- The tool shall include the ability to download a NIEM mapping spreadsheet template.  to do**

A spreadsheet can be used to define and document mappings between exchange data requirements and NIEM components.  Using the standardized NIEM mapping spreadsheet in place of a custom spreadsheet also allows this information to be imported into the tool.  The tool should make this NIEM mapping spreadsheet template easily available for download now to avoid duplication of effort in the next phase (*Map and Model*).

**Note:** The NIEM mapping spreadsheet captures three (3) primary kinds of information:

* data requirements - includes exchange data requirements like component names, data types, definitions, and cardinality
* mapping codes and notes - documents if the data requirement has a match in NIEM and provides any special notes
* NIEM components - includes the NIEM property names, and optionally the data types, definitions, and additional information

Two (2) general use cases exist for this download capability:

* For a new user accessing this phase, a blank mapping spreadsheet template will be downloaded.
* For an existing user accessing from another phase (e.g. *Map & Model*) a populated mapping spreadsheet will be downloaded.

Only the data requirements section needs to be completed during this phase.  Mappings and NIEM components will be completed during the next phase.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-118**](https://jira.di2e.net/browse/NIEM-118?src=confmacro) **- The tool shall include the ability to import data requirements from a NIEM mapping spreadsheet.  to do**

In addition to uploading the NIEM mapping spreadsheet as an artifact to include with the IEPD, the tool shall be able to parse the spreadsheet for IEPD data requirements.  These requirements will be used in the next phase, *Map and Model*.

**Note:** NIEM Mapping Spreadsheet QA functionality will also be required to ensure that the uploaded spreadsheet has the required tabs, columns, and values needed for import. The mapping spreadsheet should be 'locked' where possible to preserve format and deter users from modifying the template.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-119**](https://jira.di2e.net/browse/NIEM-119?src=confmacro) **- The tool shall include the ability to select a specific NIEM release for the IEPD.  to do**

Part of the *Analyze Requirements* phase should be to determine which NIEM release to use for the IEPD.  Developers typically choose to use the latest release of NIEM unless they need to reuse an older release of NIEM that is already  being used within their organization, by an existing EIEM or IEPD that they intend to leverage, or by their exchange partners.  Users should make this determination during requirements analysis and it should be recorded in the tool.  Future phases of the IEPD lifecycle will use this so features like NIEM component searches, component creation, and conformance validation default to the intended release.

**Note:**As identified in [_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-112](https://jira.di2e.net/browse/NIEM-112?src=confmacro) - The tool shall provide initial NIEM support beginning with the NIEM 3.0 release. **to do**, the tool will only support NIEM 3.0 onward. If there is considerable demand from the community for previous releases these will be considered for future development.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-120**](https://jira.di2e.net/browse/NIEM-120?src=confmacro) **- The tool should incorporate a Plugin Architecture to allow third-party developers to extend the ability to extract data requirements from other formats.  to do**

Earlier phases of the tool require users to extract data requirements from their own sources (UML diagrams, database schemas, etc) and translate them into the data requirements section of the mapping spreadsheet.  This feature would allow third-party developers to use the Plugin Architecture of the tool to extend capabilities of data requirements extraction to other defined formats.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-121**](https://jira.di2e.net/browse/NIEM-121?src=confmacro) **- The tool should include the ability to identify related IEPDs within a supported NIEM IEPD Registry and/or Repository.  to do**

Using some of the general IEPD metadata entered by the user in the *Scenario Planning*phase, the tool should be able to search a future NIEM IEPD Registry / Repository for related IEPDs.  During this *Analyze Requirements* phase, the IEPD developers should be able to see if there are similar IEPDs that could be reused or leveraged.

**Note:**The goal here is to leverage a proposed capability of the IEPD Registry/Repository. This feature will be dependent on an external API service to request and receive results from the IEPD Registry/Repository.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-122**](https://jira.di2e.net/browse/NIEM-122?src=confmacro) **- The tool shall include the ability to allow users to leverage a published IEPD.  to do**

If users discover an existing IEPD that meets many of their exchange requirements, the tool shall allow the IEPD to be imported and extended or otherwise modified as needed.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-123**](https://jira.di2e.net/browse/NIEM-123?src=confmacro) **- The tool may include the ability to directly enter data requirements into a web form.  to do**

In addition to uploading local data requirements via the mapping spreadsheet and potentially other file formats, users should be able to add or modify local data requirements directly in the tool via a web interface.

### Map and Model

In this stage of the IEPD Lifecycle, a user takes the data requirements identified during the *Analyze Requirements* phase, searches NIEM, records suitable matches as mappings, and identifies which requirements do not have matches and will need to be represented by local extensions.

Note that this phase is dependent on the tool's *Browse and Search* capability.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-124**](https://jira.di2e.net/browse/NIEM-124?src=confmacro) **- The tool shall include the ability to map user data requirements to NIEM data components.  to do**

The typical mapping process for users involves a lot of back-and-forth work searching and copying results between various tools and a spreadsheet – the goal of this requirement is to automate the mapping process. The current process involves:

* Capture local requirements in the mapping spreadsheet, including information like field names, data types, definitions, cardinality, classes, business rules, etc.
* For each local field, use the SSGT, Movement, release schemas, or the release documentation spreadsheet to search for a potential corresponding match in NIEM.
* Capture matches in the mapping spreadsheet, recording the mapping code and information about the NIEM component.  Information about the NIEM component namespace and name are required; additional information like definitions, data types, and code sets are sometimes also captured in order to make the mappings easier to review by subject matter experts not familiar with NIEM.
* As matches are found, users working with the SSGT may choose to go ahead and add these NIEM components to their subset at the same time, saving an extra step in the next IEPD Lifecycle phase (*Build and Validate)*

While the mapping process will always require a certain degree of grunt work to review and select the appropriate match for each data requirements, better support can still be provided.

Once local requirements are uploaded via the mapping spreadsheet in the *Analyze Requirements* phase, they should carry forward into this phase.  Users should be able to select an individual field out of this list of local requirements, optionally pre-load the field name as search terms (break up camel cased-terms in local names with spaces), search NIEM components and user extensions, and click a button to capture a new mapping.  Users may choose to map data requirements to NIEM types if no NIEM property exists with the correct semantics, but a type exists that can be reused directly, extended, or augmented.  Users may also decide that no match exists in NIEM and a custom extension will need to be created.

After the user is through with this process, an updated mapping spreadsheet should be available for download, with the original local data requirements and the newly added mapping codes and matching NIEM components, with supporting information like data types and definitions provided automatically.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-125**](https://jira.di2e.net/browse/NIEM-125?src=confmacro) **- The tool shall support a 'top-down' message development capability.  to do**

Mapping to NIEM can be done from a top-down, bottom-up, or hybrid approach.  Searching low-level fields (like first name or eye color) should lead users to larger objects they may need, like nc:PersonType (bottom-up approach).  Or first finding some of the relevant larger objects in NIEM (like nc:PersonType and nc:LocationType) may help users see the larger perspective and focus in on the right objects to select better matches (top-down approach).  Sometimes there can be more than one match that might work, but the best match can only be determined by knowing which of the larger objects are going to be used and how they relate together.

The bottom-up approach is already supported by the basic search functionality.  Results are sorted by name or namespace without a ranking or visual indicator denoting whether or not the matches are currently tied to NIEM objects already selected in the mapping.  Providing this new perspective should help users to determine the best match more easily and streamline message development.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-126**](https://jira.di2e.net/browse/NIEM-126?src=confmacro) **- The tool shall include the ability to create custom model extensions.  to do**

Users should be able to use forms to create new namespaces, properties, types, and code sets.  While users will always have the option to use their own schema editors to build schemas, a form-based user interface can simplify the learning requirements needed to be able to build NIEM content.  In addition to field names, specialized forms can provide a lot of helpful information where users actually needed it, including field descriptions, example values, required vs optional flags, representation terms and definition standard opening phrases, links to related training or videos, related NDR rules, and field validation.

Forms can also be used to walk users through some of the advanced modeling concepts in NIEM.  For example, when trying to add additional components to an existing type like nc:PersonType, a form could provide information to help guide the user between creating an augmentation, extension, or role.  Augmentation point elements could be created automatically.  Creating new code sets could be simplified:

* the same root name and definition could be used to create a code set simple type, complex type with simple content, code element, and abstract element for a substitution group head
* code values and definitions could be uploaded via a spreadsheet or copy-pasted into the UI for easy conversion into schema

**Note:**This feature will need some further consideration based on how General Users will potentially submit their custom model extensions to be (1) published by an Administrator and (2) searchable by other users.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-127**](https://jira.di2e.net/browse/NIEM-127?src=confmacro) **- The tool shall include the ability to validate NIEM conformance on individual extension components.  to do**

The tool should run conformance and QA checks as users create custom extensions.  This catches errors early on and helps users to learn NDR rules for NIEM development.

**Note:**This would not be run-time validation on keystroke, but rather validation on the web form as a whole.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-128**](https://jira.di2e.net/browse/NIEM-128?src=confmacro) **- The tool shall include the ability to migrate to a new NIEM release.  to do**

The Migration Tool is currently available to help users migrate SSGT-generated NIEM subsets from one release to the next.  This tool should support that same NIEM subset migration functionality, but also include some additional migration assistance for extensions in IEPDs and EIEMs.  Note that this tool will not ever be able to fully migrate content to a new NIEM release.  For example, if a domain removes a type, the user will need to determine if another existing type is a suitable replacement, if they need to recreate the type locally, or if the requirement has gone away and does not need to be supported any more.  However, the tool should be able of doing such things as propagating name and namespace changes across extensions.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-129**](https://jira.di2e.net/browse/NIEM-129?src=confmacro) **- The tool should leverage mappings for improved search support.  to do**

As users map local requirements to NIEM components, these local field names and definitions should be stored and used to improve NIEM searches.  NIEM has long supported a keywords field for components but this has always been very sparsely populated.  Capturing actual real-world names for NIEM components could improve search and discovery for other users.

**Note:**Users should have the ability to define whether locally added field names and definitions are *public* or *private*. If submitted for *public* usage, all field names and definitions will be reviewed and approved by the NIEM Management Office (NMO) prior to publication.

### Build and Validate

NIEM XML or NIEM JSON schemas are built during this phase of the IEPD Lifecycle.  Users start with their mapping information, which already documents which components can be found in NIEM and which will need to be created.  If it hasn't already been done during the mapping process, users can build a NIEM subset that contains only the components they need in their IEPD, plus any required dependencies.  Once a subset is built, users build additional schemas to define the extensions needed to represent data requirements not available in NIEM.

After building schemas, users should validate schemas, verify NDR conformance, and build and review sample instances as a way to ensure that the message schemas fully represent data requirements as intended.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-131**](https://jira.di2e.net/browse/NIEM-131?src=confmacro) **- The tool shall include the ability to build subsets from both NIEM releases and NIEM community content.  to do**

Mappings to NIEM components made during the *Map and Model* phase should translate into the start of a subset.  All of the designated NIEM components in the mapping should be automatically added to the IEPD subset.  Users should also be able to add additional NIEM components to the subset as well, even if they are not referenced in the mapping.

All available and conformant NIEM content that matches the user's selected NIEM release should be made available for subset.  This includes NIEM releases as well as extensions from other EIEMs and IEPDs. (**Note:** the ability to leverage community content will require interaction with the IEPD Registry/Repository *or* the ability to import an EIEM).

Subset features should include the ability for a user to:

* Add global properties and types to the subset from search results and from individual component detail pages
* Add property references to a type
* Set default and individual cardinality (min / max) on the number of allowable element repetitions within a type.
* Set default and individual nillable values.
* Remove individual properties and types from the subset list
* View the list of properties and types in the subset
* Click on properties and types in the subset list to open component detail pages

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-134**](https://jira.di2e.net/browse/NIEM-134?src=confmacro) **- The tool shall include the ability to validate NIEM Naming and Design Rules (NDR) conformance target attributes.  to do**

The NIEM Conformance Testing Assistant (ConTesA) is a separate tool that allows users to upload a schema or a zip file with multiple folders and schemas.  It looks for the conformanceTargets attribute in each schema and runs the corresponding set of NDR rules for validation.  It also generates a conformance report listing the rules that passed and failed.  This conformance report is typically included in an IEPD package as part of the conformance assertion.

ConTesA is limited to checking NDR conformance rules on NIEM XML schemas only.  The NTAC is currently reviewing the best way to represent conformance rules with the addition of (1) the NIEM metamodel as a way to describe models without the limitations of XML Schema or JSON Schema, (2) NIEM JSON as an official NIEM representation, and (3) UML as another possible representation.  While the exact way forward on this has yet to be determined, the following requirements still apply:

* The tool should generate a conformance report available for download that includes rule numbers, rule titles, file paths and line numbers for errors.
* The tool should generate a navigable list of errors and affected components so that a user may easily reach and edit the components with issues via the user interface.
* The tool should check all schemas that are not marked as external standards.  A schema with a missing conformanceTargets attribute should be identified as an error.
* Conformance validation should be available as a web service so that this and other tools may reuse its functionality.

It may be necessary to transform a NIEM model in one format to another format for conformance validation.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-135**](https://jira.di2e.net/browse/NIEM-135?src=confmacro) **- The tool shall include the ability to validate a Sample Instance (XML) against all rules within the MPD Catalog.  to do**

NIEM currently relies on users to validate sample instances against schemas for verification.  The tool shall provide both standard XML instance validation, plus validate NIEM-specific NDR instance rules**2** .  These include:

* **Rule 12-3:** An element must not have more than one attribute that is structures:id, structures:ref, or structures:uri.
* **Rule 12-6:** Check that references refer to an appropriate object (e.g., a reference for a person element should actually point to a person, an extension of, or a role of a person, and not point to something completely different like a location).
* **Rule 12-17:** Check that metadata is applied to an appropriate element or type.

The tool shall perform basic XML and NDR-specific instance validation and display validation status and errors.

### Assemble and Document

Having built schemas and sample instances during the last phase, users now assemble the files needed for the IEPD into a package and finalize the documentation and IEPD metadata.  The Model Package Description (MPD) Specification**3** provides rules and guidance on how to structure an IEPD.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-139**](https://jira.di2e.net/browse/NIEM-139?src=confmacro) **- The tool shall include the ability to review and edit IEPD metadata.  to do**

IEPD metadata that should be documented in the MPD catalog is collected during the *Scenario Planning* phase.  During the *Assemble and Document* phase, users should have a chance to review and adjust this metadata as needed.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-140**](https://jira.di2e.net/browse/NIEM-140?src=confmacro) **- The tool shall include the ability to upload and label files for the IEPD.  to do**

Any additional files that users have created for the IEPD outside of the tool should be uploaded for assembly into the IEPD.  This may include additional diagrams, readme files, master documents, schemas built via schema editors rather than in this tool, etc.  Users should label each uploaded file with the kind of IEPD artifact it is.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-141**](https://jira.di2e.net/browse/NIEM-141?src=confmacro) **- The tool shall include the ability to generate a Model Package Description (MPD) Catalog.  to do**

Using all of the available metadata, uploaded files, and tool-generated files, the tool shall construct a MPD catalog for the IEPD.  This is a required artifact which documents metadata about the IEPD and provides a file manifest for the package.  This catalog should validate against the MPD catalog schema, provided by the MPD Specification.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-142**](https://jira.di2e.net/browse/NIEM-142?src=confmacro) **- The tool shall include the ability to assemble an IEPD using the default MPD Specification package structure.  to do**

The tool shall assemble the uploaded and generated files into a package, using guidance from the MPD Specification for default folder and file names.  The tool shall make this package available for download as a zip file.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-143**](https://jira.di2e.net/browse/NIEM-143?src=confmacro) **- The tool shall include the ability to validate IEPD conformance.  to do**

Currently, verifying IEPD conformance involves uploading schemas to ConTesA for a conformance report and manually asserting that the IEPD package follows the rules specified by the NIEM [Model Package Description (MPD) Specification](https://reference.niem.gov/niem/specification/model-package-description/3.0.1/).  This reliance on user assertions for the package can lead to undetected quality issues.

The tool shall perform conformance validation for an entire IEPD package, checking:

* XML or JSON validation of schemas and sample instance instances
* NIEM NDR conformance of schemas and sample instances
* Checks for required artifacts
* Checks that paths in the MPD catalog actually exist in the IEPD
* Checks that sample instance roots match allowable message roots defined in the MPD catalog
* XML validation of the MPD catalog
* Other rules capable of automation in the MPD Specification

Results should be available to the user via the display with navigation enabled to quickly access warnings and errors, and as a report spreadsheet, with file names and line numbers where possible.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-144**](https://jira.di2e.net/browse/NIEM-144?src=confmacro) **- The tool shall include the ability to generate the IEPD conformance assertion.  to do**

As related to [_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-143](https://jira.di2e.net/browse/NIEM-143?src=confmacro) - The tool shall include the ability to validate IEPD conformance. **to do**, the tool shall be able to bundle the resulting conformance report into the IEPD package as part of the conformance assertion.  Additional information may still need to be provided for the full conformance assertion.  Users should assert which manual checks have been run, document any known issues and rationale, and provide any other such kind of information related to the review and quality of the IEPD.

### Publish and Implement

In this phase, users should publish the IEPD that has been built, making it available for reuse.  IEPDs are also turned over to developers at this point for implementation.  The detailed nature of an IEPD package is meant to provide implementers with a full set of requirements for the data exchange.

Because the tool is meant to be developed in parallel with a new NIEM IEPD Registry, functionality for this phase of the IEPD Lifecycle begins at a later phase of the tool strategy.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-145**](https://jira.di2e.net/browse/NIEM-145?src=confmacro) **- The tool shall include the ability to internally publish an IEPD or EIEM for community reuse.  to do**

The tool shall act as a NIEM content repository, allowing users to publish their own IEPDs, EIEMs, and other local models for community reuse – with explicit consent. This feature is not intended for sensitive content, and a user will need to opt-in to request publishing.  Before being accepted, contact information must be verified by the NIEM Management Office and the content must pass all applicable conformance checks.

**Note:**Given the IEPD Registry/Repository *may* require authentication/authorization to submit an IEPD and the Tool Strategy will require account management, a developer should attempt to leverage a Single Sign-On (SSO) capability to simplify the user experience across all NIEM tools.

Rather than requiring content to be centrally-vetted by NIEM governance, a badge may be provided for IEPDs or other content that undergo this additional level of manual review.

Once published, community-contributed content should be available for reuse at the IEPD level and at the individual component level.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-146**](https://jira.di2e.net/browse/NIEM-146?src=confmacro) **- The tool shall include the ability to publish IEPD metadata directly to the publicly-hosted NIEM IEPD Registry.  to do**

The tool shall use metadata from the MPD catalog to submit a new IEPD to the NIEM IEPD Registry.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-147**](https://jira.di2e.net/browse/NIEM-147?src=confmacro) **- The tool shall incorporate a Plugin Architecture to allow third-party developers to publish an IEPD package directly to a user-specified external repository.  to do**

The tool shall enable third-party developers to create plugin's that allow posting IEPDs to a user-defined external repository (e.g. GitHub).

## Browse and Search

The user should be able to browse and search NIEM release components and published community content.  The tool will provide users the ability to search and navigate through the NIEM model.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-151**](https://jira.di2e.net/browse/NIEM-151?src=confmacro) **- The tool shall provide the ability to browse NIEM release and community content.  to do**

The tool shall allow users to view information about and navigate across different contributors, models, namespaces, properties, and types, each with its own page to provide detailed information.  These pages should:

* have their own URLs to allow for bookmarks and shared links
* have collapsible sections to reduce the need for a lot of scrolling in some cases
* be customizable, so that users can easily see the kind of information that is relevant to them and hide what isn't helpful or needed

**Note:**This feature will be dependent on an external API service to request and receive results from the IEPD Registry/Repository.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-152**](https://jira.di2e.net/browse/NIEM-152?src=confmacro) **- The tool shall provide the ability to search NIEM release and community content.  to do**

The tool shall allow users to search release and community content.  The results list should provide basic information, with links to the pages containing more information.  Search functionality should include:

* Keyword searches
* Basic search operators
* Auto-completion in search box
* Results filtering
* Results sorting
* Results ranking by match quality and namespace weight (e.g., results from Core should be scored or sorted higher than results from non-vetted community contributions)

Search functionality will require a persistent data capability, like a database or SOLR.

**Note:**This feature will be dependent on an external API service to request and receive results from the IEPD Registry/Repository.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-153**](https://jira.di2e.net/browse/NIEM-153?src=confmacro) **- The tool shall include the ability to browse common NIEM components.  to do**

People who are considering or are just starting to use NIEM typically want to see what is in the model.  Their primary options now are to review dozens of schemas or large documentation files that contain information about the 11,500+ properties in NIEM.  This could be made much more user-friendly by presenting a much smaller set of widely-used components as a better entry point to the model.

A smaller list of key NIEM components will initially be a manually-curated NIEM subset.  Later on, as IEPDs are developed and community content is published, this list should be expanded to include the most frequently-used components. For instance, with a working EIEM capability, users may be able to browse local, organizational, and/or enterprise-wide custom NIEM components. A visibility filter could be used to allow users to include these custom components while browsing.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-154**](https://jira.di2e.net/browse/NIEM-154?src=confmacro) **- The tool should provide a smart search capability.  to do**

The tool should provide advanced search functionality based on NIEM-specific characteristics of the model.  For example, a user may want to search for a field within j:ArrestType or a related type.  The tool should check the type itself, but also check parent types, augmentations, extensions, and associations containing this type.  Additionally, the tool may incorporate machine learning and artificial intelligence (AI) algorithms based on user mappings to provide smarter search capabilities.

## Support Multiple Formats

The number of formats in which users interact with NIEM is growing.  Tool functionality should not be limited to a certain version of a certain representation.  Translation modules should be used for better interoperability.  The following requirements include the need to:

* Upload a file from the user
* QA the uploaded file (e.g., validation, conformance checks, checks for required columns, etc.)
* QA issue reporting
* Convert the format to an in-memory representation
* Generate the in-memory representation to the new format
* QA the output (e.g., validation)
* Download the file to the user

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-130**](https://jira.di2e.net/browse/NIEM-130?src=confmacro) **- The tool shall include the ability to load schemas and sample instances into an OpenAPI file.  to do**

OpenAPI files are commonly used to describe APIs and include endpoint descriptions, schemas, and data examples.  Schemas and examples are typically provided inline within the file.  Although the OpenAPI specification allows for references to external files, existing tools sometimes have problems with file references, both in the OpenAPI file itself and with file references between JSON schemas.

There are some open-source libraries that can be used to resolve multiple JSON schemas linked via references into a single schema, and to resolve references within an OpenAPI file to external schemas and sample instances.  This tool should take advantage of those capabilities and provide easier support to IEPD developers working with OpenAPI files:

* Provide guidance on how to add references in an OpenAPI base file to schemas and sample instances in an IEPD
* Dereference and merge multiple JSON schemas into a single JSON schema
* Dereference the OpenAPI base file (inline content from IEPD schemas and sample instances) into a separate tool-generated OpenAPI file

**Note:** Users should continue to edit the OpenAPI base file as needed and regenerate the fully dereferenced OpenAPI file to keep it in sync.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-132**](https://jira.di2e.net/browse/NIEM-132?src=confmacro) **- The tool shall include the ability to generate NIEM XML schemas.  to do**

The NIEM 4.2 release contains over 11,500 elements.  The ability to subset a release down to only the dozens or hundreds of components actually needed for a particular IEPD provides many benefits, including improved validation performance, clear limits on exactly which elements should be shared in a message, and a much smaller, custom-tailored set of files for implementers who have to review the schemas and documentation to figure out what data to send or receive.

The SSGT currently provides a way for users to select components from a NIEM release and generate only that subset, plus dependencies, into XML schemas.  This tool shall expand upon that functionality, allowing the creation of NIEM XML schemas from any NIEM model, which could include subsets of release and community content, EIEM schemas, and IEPD extension schemas.

NIEM XML schema generation requirements include:

* Generate XML schemas according to the rules defined in the NDR (schemas must pass NDR conformance checks).
* Calculate and include property, type, and namespace dependencies in the subset.
* Generate property references with user-selected cardinality
* Generate properties with user-selected nillable values
* Generate schemas with or without schema documentation.
* Inline substitution groups: Replace abstract element references in a complex type with (1) specific user-selected substitutable elements, or with (2) all available substitutable elements in the subset by default.  (New feature)

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-133**](https://jira.di2e.net/browse/NIEM-133?src=confmacro) **- The tool shall include the ability to generate NIEM JSON schemas.  to do**

Similar to the NIEM XML Schema requirement, the tool shall include the ability to generate user subsets and extensions into NIEM JSON Schemas.  The schemas should follow the NIEM JSON Specification (still in development) and guidance for the appropriate representation of components and namespaces into JSON Schema.

The generated NIEM JSON Schemas will define schemas conformant to specifications given JSON-LD as the target JSON representation.

**Note:** because the representation of NIEM content in JSON Schema is considered lossy, it is not currently recommended to provide the ability to import a NIEM JSON Schema.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-149**](https://jira.di2e.net/browse/NIEM-149?src=confmacro) **- The tool should support Java Architecture for XML Binding (JAXB) to implement IEPD packages.  to do**

The tool should use JAXB to produce automated mappings between NIEM XML schemas in the IEPD and Java objects.   This makes it simpler for Java implementers to read and write XML messages.  NIEM already provides an [IEPD Java Bindings](https://github.com/NIEM/IEPD-Java-Bindings)**4**library to support this capability - this should be integrated directly into the tool for easier usage.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-150**](https://jira.di2e.net/browse/NIEM-150?src=confmacro) **- The tool should incorporate a Plugin Architecture to allow third-party developers to convert schemas into other programming language-specific modules.  to do**

Similar to providing JAXB support for Java developers, the tool should provide common language-specific representations of IEPD schemas to simplify and improve the consistency of IEPD implementations. This feature would allow third-party developers to use the Plugin Architecture of the tool to extend capabilities of converting schemas into other programming languages (e.g. JavaScript, Python).

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-155**](https://jira.di2e.net/browse/NIEM-155?src=confmacro) **- The tool shall include the ability to import and export a NIEM model conforming to the NIEM metamodel.  to do**

The NIEM metamodel is a fully-specified representation of NIEM content, not limited by what can be defined within an XML or JSON schema.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-156**](https://jira.di2e.net/browse/NIEM-156?src=confmacro) **- The tool shall include the ability to import and export a NIEM mapping spreadsheet.  to do**

The NIEM mapping spreadsheet allows users to define data requirements and mappings to corresponding components in NIEM.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-158**](https://jira.di2e.net/browse/NIEM-158?src=confmacro) **- The tool shall include the ability to import and export NIEM XML schemas.  to do**

NIEM will soon have six different representations of NIEM XML Schemas:

* NIEM 1.0
* NIEM 2.0
* NIEM 2.1
* NIEM 3.x
* NIEM 4.x
* NIEM 5.x

The tool shall provide the ability to read and generate NIEM XML Schemas starting with NIEM 3.0-based schemas (NDR 3.0).  The tool shall follow the appropriate NIEM NDR for the syntax.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-159**](https://jira.di2e.net/browse/NIEM-159?src=confmacro) **- The tool should include the ability to import and export UML.  to do**

The tool shall use a lightweight profile of UML to enable a NIEM UML representation that is capable of staying in sync with the NIEM releases.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-160**](https://jira.di2e.net/browse/NIEM-160?src=confmacro) **- The tool should include the ability to export documentation spreadsheets.  to do**

The tool shall generate a spreadsheet as a data dictionary-like representation for NIEM model content.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-161**](https://jira.di2e.net/browse/NIEM-161?src=confmacro) **- The tool should include the ability to export interactive diagrams.  to do**

The tool shall generate interactive diagrams that allow users to view and navigate around related components.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-162**](https://jira.di2e.net/browse/NIEM-162?src=confmacro) **- The tool should incorporate support for Google Protocol Buffers.  to do**

The tool should generate NIEM mappings to Google Protocol Buffers**5**.

*"Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data – think XML, but smaller, faster, and simpler. You define how you want your data to be structured once, then you can use special generated source code to easily write and read your structured data to and from a variety of data streams and using a variety of languages." –*[*https://developers.google.com/protocol-buffers*](https://developers.google.com/protocol-buffers)

## Interfaces

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-163**](https://jira.di2e.net/browse/NIEM-163?src=confmacro) **- The tool shall include a user interface (UI) for the format translator API.  to do**

Users should be able to use the tool simply to translate between different representations of a NIEM model.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-164**](https://jira.di2e.net/browse/NIEM-164?src=confmacro) **- The tool shall include a user interface for NIEM conformance validation.  to do**

Users should be able to use this tool to check NIEM conformance on uploaded artifacts, without walking through the guided IEPD development process.  Validation requirements include:

* Validate and test NDR conformance on NIEM XML Schemas.
* Validate and test NDR conformance on NIEM JSON Schemas.
* Validate and test NDR and IEPD conformance on sample instances
* Test IEPD conformance based on MPD Specification rules on an IEPD

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-165**](https://jira.di2e.net/browse/NIEM-165?src=confmacro) **- The tool shall include a user interface to search and browse NIEM content.  to do**

Users should be able to use this tool to browse NIEM release and community content without going through the process of building an IEPD or a subset or mapping to NIEM.

## Help System

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-166**](https://jira.di2e.net/browse/NIEM-166?src=confmacro) **- The tool shall incorporate functional user interface (UI) elements that provide assistance/guidance for each software feature.  to do**

Help information should be available on each page, explaining NIEM concepts and tool support as needed.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-167**](https://jira.di2e.net/browse/NIEM-167?src=confmacro) **- The tool shall incorporate any relevant technical training material that is publicly available.  to do**

The tool should provide information or links to training materials (e.g., [niem.github.io](https://niem.github.io)) or videos where relevant.  This will help decrease the initial NIEM learning curve and help users learn about concepts and rules as they are actually building NIEM content.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-168**](https://jira.di2e.net/browse/NIEM-168?src=confmacro) **- The tool shall include a fully functional, NIEM-conformant IEPD example for demonstration purposes.  to do**

One of the IEPDs published in the tool shall be a specially-selected or created IEPD demonstrating good NIEM practices.  This demo IEPD shall be made easily accessible for user reference.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-169**](https://jira.di2e.net/browse/NIEM-169?src=confmacro) **- The tool shall include an end-to-end tutorial that explains how to use core software features.  to do**

The tool shall include written instructions and screenshots, animated GIF's, and/or video walk-throughs demonstrating how to accomplish key tasks in NIEM (e.g., build an IEPD or check NIEM conformance of an artifact).

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-170**](https://jira.di2e.net/browse/NIEM-170?src=confmacro) **- The tool shall include a Contact page.  to do**

The tool shall include instructions on how to contact the NIEM Management Office (NMO) for assistance.

## Backlog

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-136**](https://jira.di2e.net/browse/NIEM-136?src=confmacro) **- The tool shall include the ability to validate a Sample Instance (JSON) against an MPD Catalog.  to do**

As with XML instances, NIEM currently relies on users to verify that sample JSON instances validate against the JSON schemas.  The tool shall perform this validation and display validation status and errors.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-137**](https://jira.di2e.net/browse/NIEM-137?src=confmacro) **- The tool should include the ability to validate NIEM Naming and Design Rules (NDR) conformance for pre-NIEM 3.0 schemas.  to do**

The NIEM 3.0 release came with several major changes.  It introduced the conformanceTargets attribute, which NIEM uses to indicate the applicable set of NDR conformance rules to use for NDR conformance validation.  NIEM 3.0 changed the way schemas were represented, including the introduction of augmentations and the removal of appinfo in component annotations.  And the NDR was restructured to provide actual Schematron representations of rules where possible to allow for easy, consistent conformance checking.  The implementation of conformance validation for schemas prior to NIEM 3.0 may require additional effort to enable but may be important for legacy support.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-138**](https://jira.di2e.net/browse/NIEM-138?src=confmacro) **- The tool shall include the ability to build a Sample Instance.  to do**

Common XML editors, like Oxygen and XML Spy, currently provide the ability to create sample instances based on provided schemas.  The actual values used in these kinds of sample instances are typically very generic (e.g., <nc:PersonGivenName>string</nc:PersonGivenName>).  This tool should provide the ability to generate sample instances from the schemas, leveraging user-provided sample values in the data requirements section of the mapping spreadsheet.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-172**](https://jira.di2e.net/browse/NIEM-172?src=confmacro) **- The tool may include the ability to capture exchange data requirements in a custom mapping spreadsheet.  to do**

This feature would allow users to upload a custom spreadsheet that contains local data requirements rather than using the mapping template.  The tool would likely ask users to map their columns to a list of expected fields and would then extract the data.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-173**](https://jira.di2e.net/browse/NIEM-173?src=confmacro) **- The tool may include the ability to allow a user to document business rules.  to do**

Business rules, including data constraints that cannot be easily represented in NIEM schemas, are important to capture in an IEPD.  The tool may provide a consistent way to allow users to document these business rules.  Additionally, the tool may include some support for providing a form interface for generating common business rules into such things as Schematron or JavaScript functions for automated checking of business rules outside of XML instance validation.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-174**](https://jira.di2e.net/browse/NIEM-174?src=confmacro) **- The tool may include the ability to identify Communities of Interest (COIs) and exchange partners.  to do**

**Note:**This would not be a user matching feature. A user would be expected to select existing or add COI/exchange identifiers when an IEPD is published in order to generate this list.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-175**](https://jira.di2e.net/browse/NIEM-175?src=confmacro) **- The tool may include the ability to create additional exchange artifacts.  to do**

There may be additional IEPD artifacts that are commonly used, especially within certain communities.  For example, within the NIEM MilOps community, IEPDs contain a distribution statement.  This tool may provide support for commonly used community artifacts such as these.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-176**](https://jira.di2e.net/browse/NIEM-176?src=confmacro) **- The tool may include the ability to support custom package layouts.  to do**

Initially, the tool shall use the recommended IEPD directory layout from the MPD Specification for assembly.  This feature would allow users to customize folder layouts and file names.  These changes would also need to be reflected in the file manifest portion of the MPD catalog.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-177**](https://jira.di2e.net/browse/NIEM-177?src=confmacro) **- The tool should include the ability to import and export a SSGT wantlist.  to do**

The SSGT uses wantlist files to describe the contents of a NIEM release subset.  The tool should be able to read and generate this format if there is demand to let users jump back and forth between the SSGT and this tool.

[**_scroll_external/other/viewavatar-avatarid-13933-avatartype-issuetype-size-xsmall-7660f5f9d081dbd6ac2bcef9132e8d3976e8658945b63cdbb5d1a2c1c306922eNIEM-178**](https://jira.di2e.net/browse/NIEM-178?src=confmacro) **- The tool should interact with the SSGT to provide NIEM JSON Schema subsets from a wantlist.  to do**

The tool shall include a webhook (or endpoint) for the Format Translator API that the Schema Subset Generation Tool (SSGT) can leverage to return a NIEM-JSON Schema Subset.

# Non-Functional Requirements

A non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system or the constraints under which the entire system must operate (such as performance requirements, security, or reliability) rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions.**6**

|  |  |
| --- | --- |
| **NFR Category** | **Requirements** |
| **Audit** | * The system should incorporate an audit capability so all create, read, update, and delete (CRUD) operations are stored in an audit database. |
| **Accessibility** | * The system shall be functional on all industry-standard browsers including Microsoft Edge, Mozilla Firefox, and Google Chrome. |
| **Availability** | * The system shall be operational no less than 99% of the time. * The system shall include maintenance procedures as part of the system documentation. * The system shall include restore and reactivate procedures as part of the system documentation. |
| **Backup and Restore** | * The primary system administrator, or designated administrator, should be responsible for all backup and restore operations. * A full backup should be completed weekly with a daily incremental backup performed between weekly backups. |
| **Capacity** | * The memory and storage requirements should be part of the system documentation listed as minimum system requirements. |
| **Certification and Compliance** | * The system shall comply with IT Accessibility Laws and Policies ADA Section 508**7**. |
| **Documentation** | The system shall include standard system documentation including, but not limited to:   * User Guides * Online Help capability * Source Code * Data Model * Technical Manuals |
| **Efficiency** | * The system should notify the user of an error with information describing the error in a non-technical manner where possible. * The system should record all errors and store them in a database for future analysis. |
| **Interoperability** | * TBD (need to assess IEPD Registry/Repository integration considerations) |
| **Maintainability** | * The system shall include maintainability procedures as part of the system documentation. |
| **Modifiability** | * The system shall incorporate free open-source development framework and libraries where possible. * The system shall allow localization of language variables in case developers would like to translate for external use.   + The system shall be designed to support the English-speaking user in the United States (en\_US.UTF-8 locale). |
| **Privacy** | * TBD |
| **Procedures** | * TBD |
| **Redundancy** | * TBD |
| **Resource Management** | * The GUI, tool tips and Help Pages should be sufficient for the User to interact successfully without formal training. YouTube videos demonstrating key system capabilities would be useful with a link to the Tool Strategy Help Pages. |
| **Security (RFM)** | * TBD |
| **Performance** | * TBD |
| **Portability** | * The system should be portable. So moving from one OS to other OS does not create any problem. |

# Use Cases

## UC1

*TBD*

### Purpose

### Actors

### Pre conditions

### Post conditions

### Main Flow

### Exceptions

# Next Steps

* Conduct a 'crosswalk' between potential points of integration between the Tool Strategy and IEPD Registry/Repository applications.

# References

1. Validating XML with Schematron, XML.com - <https://www.xml.com/pub/a/2000/11/22/schematron.html>
2. National Information Exchange Model Naming and Design Rules, Version 4.0 - <https://reference.niem.gov/niem/specification/naming-and-design-rules/4.0/niem-ndr-4.0.html>
3. National Information Exchange Model Model Package Description Specification, Version 3.0.1 - <https://reference.niem.gov/niem/specification/model-paakage-description/3.0.1/model-package-description-3.0.1.html>
4. NIEM Java Bindings, Github Repository - <https://github.com/NIEM/IEPD-Java-Bindings>
5. Google Protocol Buffers, Google Developers Guide - <https://developers.google.com/protocol-buffers>
6. Adams, K.M. (2015). "3.2 Definitions for Functional and Non-Functional Requirements". Non-functional Requirements in Systems Analysis and Design. Springer. pp. 45–50.
7. ADA Section 508 - [https://www.section508.gov/manage/laws](https://www.section508.gov/manage/laws-and-)

# Appendix

## NBAC Annual Report - Tool Recommendations

* Provide an end-to-end tool capability to facilitate and automate all phases of the NIEM IEPD Lifecycle, including tools to trace information exchange requirements from architecture scenario planning artifacts to mapping extract, transform, load (ETL) operations, IEPD implementation, and IEPD posting for discovery and reuse.
* Provide the Community with an Enterprise Architecture Tool to document “as is” and “to-be” architectures and exchange models.
* Integrate existing Movement and SSGT tools.
* NIEM should position itself as an information exchange 'translator' between standards to alleviate it being dismissed as a competing or standard.
* Consider alternate representations of NIEM XML (e.g. JSON, UML, etc.) to make it easier for tool developers to work within the NIEM framework.
* Provide comprehensive NIEM JSON tool support.
* Allow SSGT to restrict data type facets in subset schema generation. For example, string minimum Length or integer minimum Inclusive values.
* Provide an easy browsing tool for NIEM core and Domain content.

## Requirements Discussion (GTRI Site Visit)

The 73 Functional Requirements in the current version of this document were generated from internal discussion between representatives of the NIEM Management Office (NMO) and Lead Developer (GTRI) on January 21-23, 2020 in Atlanta, GA. The team reviewed some resources provided by the former Management Office as well as user feedback with the NIEM community over the past several years.

The goal of these meetings was to:

* Review the current NIEM Tool Catalog and assess capabilities
  + Identify strengths and weaknesses of the:
    - User Interface
    - Tool Architecture
    - Workflow
* Conduct gap analysis of tool support for the IEPD Lifecycle Process
* Identify high-level “features” or “groups” for organization of requirements
* Discuss risks, assumptions, and limitations of the Tool Strategy
* Assess ROI of requirements and dependencies (as needed)

A Spreadsheet was generated to identify a proposed order of operations, level of effort (LOE), and three (3) phased approach as briefed at the October 2019 NIEM Face-to-Face (F2F) Meeting. For purposes of this document we are not including this information as it is speculative and will likely change based on user feedback, resource availability, and funding options available.

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Analyze Requirements** | The second phase of the IEPD Lifecycle. The purpose of the Analyze Requirements phase is to further elaborate the selected information exchange in order to understand and document the business context and data requirements. The deliverables for this phase are Business Requirements and Business Rules. |
| **Augmentation** | A NIEM technique that enables the reuse of type extensions that occur within particular domains for use elsewhere. Type augmentation avoids the need to create new specialized entities and duplicate type extensions that could not have been reused. Instead, the technique simply supplements an existing type with a reusable set of properties required for a given context. |
| **Build and Validate** | The fourth phase of the IEPD Lifecycle. The purpose of the Build and Validate phase is to create a set of exchange-specific NIEM-conformant XML schemas that form the data model created for the exchange. The deliverables for this phase are NIEM XML Schemas and the XML Wantlist. |
| **Codelist** | NIEM XMLCodelists are used to constrain possible values for a data element to a predetermined, NIEM-conformant list to preserve integrity. |
| **Conformance** | The requirement that those who participate in NIEM by contributing data components or creating and sharing IEPD artifacts are following the agreed-upon procedures for doing so, and that all documentation meets minimum criteria and the NIEM Naming and Design Rules where applicable. |
| **Conformance Schema** | A schema that maintains the XML Schema syntax requirements of NIEM as specified by the NIEM Naming and Design Rules. |
| **Conformance Validation Tool** | The NIEM Conformance Tool is an option available for you to validate schema. The Conformance Validation Tool is a dynamic tool that allows you to upload an IEPD, a set of schemas in a zip file, or an individual schema. |
| **Element** | The fundamental building block of an XML document. XML elements can contain other elements or content. XML elements are composed of a start tag, content, and end tag and may include attributes. |
| **Exchange Mapping** | The process of comparing desired exchange content with the exchange specifications in order to ensure semantic compatibility prior to information exchange. |
| **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 Naming and Design Rules (NDR). |
| **Lossy** | Refers to when file compression results in lost data and quality from the original version. |
| **Map and Model** | The third phase of the IEPD Lifecycle. The purpose of the Map and Model phase is to associate local objects with types and elements in NIEM. This process is called mapping a domain model to NIEM. The deliverables for this phase are Exchange Content Model and Mapping Document. |
| **Mapping Document** | The Mapping Document is a tool used to bridge the gap between exchange data and the data objects reused from the NIEM data model. The Mapping Document is a standardized document used to identify how exchange data aligns to, or “maps,” and reuses NIEM data objects. The Mapping Document is a local ‘data dictionary’ of NIEM XML data objects, local exchange data objects, and the relationship between the two. |
| **Message** | In an information exchange messages contain objects that are instances generated from the schema by the sender of the message. At any time, if needed, instances may be validated against the schema by either the sender or receiver of the message. |
| **Objects** | Constructs that represents a person, place, thing, etc. |
| **Protocol Buffers** | Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data – think XML, but smaller, faster, and simpler. You define how you want your data to be structured once, then you can use special generated source code to easily write and read your structured data to and from a variety of data streams and using a variety of languages. |
| **Publish and Implement** | The last phase of the IEPD Lifecycle. The purpose of the Publish and Implement phase is to publish IEPD for search, discovery, and reuse. The deliverables for this phase is the publication of IEPD to online repository and to implement the exchange. There are no required artifacts for this phase. |
| **Reconciliation** | The process of bringing two differing data sets or processes together to be synchronized in order to promote interoperability between them. |
| **Reference IEPD** | An IEPD that has been designated as a reference IEPD has been endorsed by an Authoritative Source as a shining example or base exchange template that should be reused as-is or modified to perform a similar business function. |
| **Reference Schema** | A schema that is intended to be the authoritative definition schema for a NIEM namespace. This includes the reference schemas for the NIEM Core schema and NIEM domain schemas. |
| **Registry** | Authoritative, centrally controlled store of information that facilitates discovery and reuse. A NIEM registry of IEPDs would act as a store or pointer to all known IEPDs in existence or currently under development to allow implementers to take advantage of parallel efforts. |
| **Release** | A set of schemas published by the NIEM Program Management Office (PMO) at <http://release.niem.gov/niem/> and assigned a unique version number. Each schema defines data components for use in NIEM information exchanges. |
| **Repository** | An information system used to store and access information, schemas, stylesheets, controlled vocabularies, dictionaries, and other work products. It would normally be discovered via a registry. |
| **Scenario Planning** | The first step of the IEPD Lifecycle. Existing documentation should be reviewed to determine such things as the stakeholders of the exchange, current technical architecture requirements, security, privacy, and other-related concerns, content requirements, and the use of external standards. |
| **Schematron** | Schematron is an XML schema language, and it can be used to validate XML. Oxygen XML Editor is a commercial tool that provides a Schematron-validation capability that can be used to test NDR conformance of XML schemas. |
| **Type** | A description of a class of objects that share the same operations, abstract attributes and relationships, and semantics. The operations aspect of a type is a programming concept related to methods and is, therefore, not applicable in NIEM, which uses only the data aspects. |
| **Type Extension** | A description of a class of objects that share the same operations, abstract attributes and relationships, and semantics. The operations aspect of a type extension is a programming concept related to methods and is, therefore, not applicable in NIEM, which uses only the data aspects. |
| **Validation** | The documented process of showing that a system is stable and capable of producing pre-determined outcomes; Answers the question of whether or not it does what the user really requires. |
| **Wantlist** | A portable construct used in the SSGT to save and reuse schema subsets of the overall NIEM data model. A want list can be saved or loaded directly from the SSGT tool. A want list is an XML instance that specifies the NIEM data components required (and therefore selected) by the user for the subset schema he/she is building. It does not include NIEM data components the user‐selected set depends on and therefore, will appear in the subset for the want list. |

# Acronyms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **ADA** | Americans with Disabilities Act of 1990 |
| **API** | Application Programming Interface |
| **COI** | Community of Interest |
| **CRUD** | Create, Read, Update and Delete |
| **ConTesA** | Conformance Testing Assistant |
| **DOD** | Department of Defense |
| **EIEM** | Enterprise Information Exchange Model |
| **ETL** | Extract, Transform, Load |
| **F2F** | Face-to-Face |
| **GIF** | Graphics Interchange Format |
| **GTRI** | Georgia Tech Research Institute |
| **IEPD** | Information Exchange Package Documentation |
| **IEPD** | Information Exchange Package Documentation Lifecycle |
| **JAXB** | Java Architecture for XML Binding |
| **JSON** | JavaScript Object Notation |
| **JSON-LD** | JavaScript Object Notation for Linked Data |
| **LOE** | Level of Effort |
| **MilOps** | Military Operations Domain |
| **MPD** | Model Package Description |
| **MVP** | Minimum Viable Product |
| **NBAC** | NIEM Business Architecture Committee |
| **NDR** | Naming and Design Rules |
| **NFR** | Non-Functional Requirement |
| **NIEM** | National Information Exchange Model |
| **NMO** | NIEM Management Office |
| **NTAC** | NIEM Technical Architecture Committee |
| **OAS** | OpenAPI Specification |
| **RBAC** | Role-Based Access Control |
| **ROI** | Return of Investment |
| **ROM** | Rough Order of Magnitude |
| **SOLR** | Searching On Lucene w/Replication |
| **SSGT** | Schema Subset Generation Tool |
| **SSO** | Single Sign-On |
| **SQL** | Structured Query Language |
| **TS** | Tool Strategy |
| **UI** | User Interface |
| **UML** | Unified Modeling Language |
| **URI** | Uniform Resource Identifier |
| **QA** | Quality Assurance |
| **XML** | Extensible Markup Language |