



NIEM Conformance

NIEM Technical Architecture Committee (NTAC)

**15 September 2008
Version 1.0**

Change History

No.	Date	Reference: All, Page, Table, Figure, Paragraph	A = Add. M = Mod. D = Del.	Revised By	Change Description
1.0	08/07/2008	All	A	NTAC	Initial draft version

Contents

1	Purpose	1
2	Definition of NIEM <i>conformance</i>	1
3	NIEM does not define <i>compliance</i>	1
4	NIEM-aware or -supporting vs. NIEM-conforming.....	2
5	Detailed information sources for NIEM conformance	2
6	Benefits of IEPDs that conform to NIEM	3
7	Suggestions to IEPD developers.....	3
8	Tools for NIEM conformance testing or validation	4

1 Purpose

This document provides a normative definition for the meaning of NIEM conformance, what it applies to, and what it does not apply to. It also outlines the benefits of conformance, and suggests sources of information, tools, and help for learning about NIEM conformance and its application.

2 Definition of NIEM *conformance*

NIEM only defines three types of *conformance*:

1. NIEM XML schemas conform to the NIEM Naming and Design Rules (NDR).
2. NIEM XML instances conform by correctly validating to NIEM-conforming XML schemas, with additional conformance rules specified by the NIEM NDR.
3. NIEM Information Exchange Package Documentation (IEPD) conforms to the NIEM IEPD Specification, which requires that their XML schemas and instances are NIEM-conforming.

The NIEM PMO specifies that an IEPD conforms to NIEM under the following conditions:

1. Each XML schema within the IEPD adheres to all rules in the [NIEM NDR](#) for this schema's conformance target class (i.e., subset schema rules, extension schema rules, exchange schema rules, or constraint schema rules).
2. Each XML sample instance within the IEPD adheres to all rules in the NIEM NDR for XML instances.
3. The IEPD itself adheres to the [NIEM IEPD Specification](#) (including required files, packaging, metadata, etc.). This requires its schemas and sample instances conform to the NIEM NDR.
4. If an existing NIEM component matches the business semantics required by the IEPD, then that component is used by the IEPD, either directly or as the basis for derived components. That is, the IEPD does not unnecessarily duplicate NIEM components.
5. Each NIEM component that is used by the IEPD, either directly or as the basis for derived components, is used in a manner consistent with the component's structural definition and business semantics. That is, the IEPD preserves semantic and structural consistency.

There are subjective factors in applying these rules that require diligent consideration by the organization(s) developing the IEPD. It is important that decisions about semantics be made by (or reviewed by) business subject-matter experts who have thorough knowledge of the business processes and domain in which the exchange is involved.

3 NIEM does not define *compliance*

NIEM does NOT define *compliance* because the term implies enforcement and the existence of an official certification process that verifies conformance or level of conformance. A formal certification process is envisioned, but does not yet exist. Therefore, *compliance* to NIEM is currently undefined and has no meaning.

Early in the development of the Global Justice XML Data Model (GJXDM) and NIEM, governance committees discussed grant language related to use of GJXDM and NIEM, and considered both terms. They concluded that the term *compliance* does not convey appropriate intent; it is too regulatory (for example, we have to comply with the U.S. tax code under fear of punishment). The objectives of GJXDM and NIEM are to develop exchange standards users want to use, but are not forced to use. Furthermore, the purpose of both programs is to facilitate and support interoperability, NOT to force or regulate it.

Therefore, the appropriate assertion is the following: Grantees comply with grant language, and that language requires that NIEM IEPDs, XML schemas, and instances, conform to NIEM. (See [NIEM Implementation Guidelines](#); these guidelines also contain general conformance rules.)

4 NIEM-aware or -supporting vs. NIEM-conforming

Systems, tools, and databases DO NOT and CAN NOT conform to NIEM. Only XML schemas, XML instances, and IEPDs can conform to NIEM. For any other entity or artifact, NIEM conformance is undefined. Internal names, or usage of data, within a given system, tool, or database have absolutely no impact on determination of NIEM conformance. Conformance is only about the format of payload data encapsulated in XML instances that validate to XML schemas that adhere to the NIEM NDR.

A non-NIEM XML schema that copies, maps to, or uses the component names of NIEM XML schemas or components without importing NIEM namespaces does not conform to NIEM. Conforming to NIEM requires that exchange schemas reuse NIEM reference schemas (or subsets of) by importing (xsd:import) NIEM schemas (or valid subsets of NIEM schemas) that define NIEM namespaces that compose the NIEM releases.

A tool, system, or database may have capabilities that specifically support the development of NIEM-conforming IEPDs, or it may be part of a development environment that specifically supports the implementation of NIEM-conforming IEPDs, or it may be used to test or verify NIEM-conformance, etc. Also, a system may provide the capability to generate, send and receive, and/or process NIEM-conforming exchanges. Such tools or systems are NOT considered NIEM-conforming, however, since they do support conformance, they are considered NIEM-aware or NIEM-supporting.

NIEM allows various extension methods to create additional XML components to use data that is not otherwise defined in NIEM. Tools that assist in creation of such extensions are also considered NIEM-aware or –supporting as well. However, implementers should be careful to avoid violating general conformance rules regarding duplication and consistency outlined in [NIEM Implementation Guidelines](#).

5 Detailed information sources for NIEM conformance

[NIEM Implementation Guidelines](#) describes NIEM conformance for grants.

NIEM conformance means adherence to the following specifications:

- [NIEM Naming and Design Rules \(NDR\) v1.2](#) – the current normative reference for XML schema conformance to NIEM.
- [NIEM IEPD Specification v2.1](#) – the current normative reference for NIEM IEPDs.

Another useful document is [Techniques for Building and Extending NIEM XML Components v2.0.1](#) – a practical guide for NIEM (a near-term substitute for the User Guide that is still in progress).

6 Benefits of IEPDs that conform to NIEM

There are a number of advantages to ensuring that an IEPD conforms to NIEM, including:

- NIEM-conforming IEPDs are easier to work with now and in the future – which promotes reuse. One reason is because their schemas use a profile of XML Schema, thus limiting the number of acceptable XML Schema mechanisms that can be used.
- NIEM-conformant IEPDs can be built, modified, stored, shared, and reused with NIEM-aware reference tools (at <http://niem.gov>), and with other current and future tools that follow the NIEM Tool Architecture under development by the NTAC.
- NIEM-conforming IEPDs can have a higher degree of semantic and structural data interoperability.
- Migration support exists (again through reference tools) for NIEM-conforming IEPDs when they need to be updated to newer-release versions of NIEM.
- NIEM-conforming IEPDs can be registered in the NIEM IEPD Registry/Repository and socialized in the [NIEM IEPD Clearinghouse](#).
- Extension schemas in registered NIEM-conformant IEPDs may be reviewed by governance bodies for potential additions to future releases – a simple way to get local requirements into NIEM.
- Additional tool support for registered NIEM-conformant IEPDs is planned.
Example: Generating impact reports that would show how an upgrade to the latest version of NIEM could affect an IEPD (e.g., within this IEPD, this NIEM component has been dropped, these two have been renamed, this one has moved from domain X to NIEM Core, etc.).

7 Suggestions to IEPD developers

Obviously, being as knowledgeable as possible with NIEM and its documentation is important. To help increase understanding and improve their ability to build NIEM-conforming IEPDs, developers should:

- Use NIEM-aware reference tools that NIEM PMO provides at <http://niem.gov>, or other current and future tools that follow the NIEM Tool Architecture under development by the NTAC.
- Use methods from other NIEM-conforming schemas (e.g., match templates of existing schemas)
- Adhere to the [NIEM NDR](#) and [IEPD Specification](#).
- Use [Techniques for Building and Extending NIEM XML Components v2.0.1](#) – a near term substitute for the User Guide.

-
- Follow guidelines presented in the NIEM training and in the User Guide (not yet available).

8 Tools for NIEM conformance testing or validation

Many tools exist for automatically validating that an XML instance accurately corresponds to an XML Schema definition. See <http://www.w3.org/XML/Schema.html> or <http://www.w3.org/2001/03/webdata/xsv>.

No tool exists to automatically ensure 100% NIEM conformance of schemas and IEPDs, because some NIEM criteria require subjective evaluation. For example, a tool cannot automatically verify that the definition of a NIEM component is constructed per ISO Standard 11179, and is semantically accurate.

However, the NIEM Conformance Tool can review XML schemas and IEPDs and automatically identify some violations of the NDR and the IEPD Specification that confirm non-conformance. The NIEM Conformance Tool is in progress and will be released for public use soon after the approval and public release of NIEM NDR 1.3 (nearing completion). The tool has been demonstrated to the NIEM governance bodies and was well-received. It will provide a simple but fairly comprehensive online facility to automatically validate schema conformance against the NIEM NDR (through Schematron rules that represent the NDR), and IEPD conformance against the current NIEM IEPD Specification. The tool provides a report of both automatic and manual conformance checks. The report is editable so that as each manual check is performed, it can be marked as a pass or fail.