UML Profile for NIEM 3 (NIEM-UML-3)

*Initial Submission*

**OMG Document Number:** gov/2014-08-02

**Machine Consumable Files:**

*Note: work-in-progress machine-readable files may be found at* [*https://github.com/NIEM/NIEM-UML/tree/master/Specification*](https://github.com/NIEM/NIEM-UML/tree/master/Specification)

Normative:

Non-normative:

Copyright © 2014 Data Access Technologies (Model Driven Solutions)

Copyright © 2012 Georgia Tech Research Institute (GTRI)

Copyright © 2012 Microsoft

Copyright © 2014 Object Management Group (OMG)

Copyright © 2012 Visumpoint

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty- free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems-- without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING

LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227- 7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 140 Kendrick Street, Needham, MA 02494, U.S.A.

TRADEMARKS

MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™ , Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IIOP™ , IMMT™, MOF™, OMG Interface Definition Language (IDL) ™, and OMG Systems Modeling Language (OMG SysML) ™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials. Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

Table of Contents

[0 Submission-related material 0-I](#_Toc406072851)

[0.1 Submission Introduction 0-I](#_Toc406072852)

[0.2 Submission Team 0-I](#_Toc406072853)

[0.2.1 Submitters 0-I](#_Toc406072854)

[0.2.2 Government Stakeholders 0-I](#_Toc406072855)

[0.2.3 Contributors 0-I](#_Toc406072856)

[0.3 Resolution of Requirements 0-I](#_Toc406072857)

[0.3.1 Mandatory requirements 0-I](#_Toc406072858)

[0.3.2 Non-mandatory features 0-II](#_Toc406072859)

[0.4 Resolution of Discussion Issues 0-II](#_Toc406072860)

[Preface 3](#_Toc406072861)

[1 Scope 5](#_Toc406072862)

[2 Conformance 6](#_Toc406072863)

[3 Normative References 7](#_Toc406072864)

[4 Terms and Definitions 8](#_Toc406072865)

[5 Symbols 9](#_Toc406072866)

[6 Additional Information 10](#_Toc406072867)

[7 NIEM-UML Modeling Guide 11](#_Toc406072868)

[8 NIEM-UML Profile Reference 12](#_Toc406072869)

[8.1 Profile : Model\_Package\_Description\_Profile 12](#_Toc406072870)

[8.1.1 Overview 12](#_Toc406072871)

[8.1.2 <Stereotype> ApplicationInfo 12](#_Toc406072872)

[8.1.3 <Stereotype> BusinessRulesArtifact 12](#_Toc406072873)

[8.1.4 <Stereotype> ChangeInformationType 12](#_Toc406072874)

[8.1.5 <Stereotype> ChangeLogType 13](#_Toc406072875)

[8.1.6 <Stereotype> ConformanceAssertion 13](#_Toc406072876)

[8.1.7 <Stereotype> ConformanceReport 14](#_Toc406072877)

[8.1.8 <Stereotype> Documentation 14](#_Toc406072878)

[8.1.9 <Stereotype> ExtensionSchemaDocument 14](#_Toc406072879)

[8.1.10 <Stereotype> ExternalSchemaDocument 14](#_Toc406072880)

[8.1.11 <Stereotype> File 14](#_Toc406072881)

[8.1.12 <Stereotype> FileType 15](#_Toc406072882)

[8.1.13 <Stereotype> IEPSampleXMLDocument 15](#_Toc406072883)

[8.1.14 <Stereotype> ModelPackageDescriptionRelationship 15](#_Toc406072884)

[8.1.15 <Stereotype> MPDChangeLog 16](#_Toc406072885)

[8.1.16 <Stereotype> ReadMe 16](#_Toc406072886)

[8.1.17 <Stereotype> ReferenceSchemaDocument 16](#_Toc406072887)

[8.1.18 <Stereotype> RelaxNGSchema 16](#_Toc406072888)

[8.1.19 <Stereotype> RequiredFile 17](#_Toc406072889)

[8.1.20 <Stereotype> SchematronSchema 17](#_Toc406072890)

[8.1.21 <Stereotype> SubsetSchemaDocument 17](#_Toc406072891)

[8.1.22 <Stereotype> Wantlist 17](#_Toc406072892)

[8.1.23 <Stereotype> XMLCatalog 17](#_Toc406072893)

[8.1.24 <Stereotype> XMLSchemaDocument 18](#_Toc406072894)

[8.1.25 <Artifact> ArtifactOrArtifactSet 18](#_Toc406072895)

[8.1.26 <Artifact> ConformanceTargetType 18](#_Toc406072896)

[8.1.27 <Artifact> ConstraintSchemaDocumentSet 18](#_Toc406072897)

[8.1.28 <Artifact> ContactInformationType 18](#_Toc406072898)

[8.1.29 <Artifact> DescribedType 19](#_Toc406072899)

[8.1.30 <Artifact> EntityRepresentation 19](#_Toc406072900)

[8.1.31 <Artifact> EXIXMLSchemaType 19](#_Toc406072901)

[8.1.32 <Artifact> FileSet 20](#_Toc406072902)

[8.1.33 <Artifact> FileSetType 20](#_Toc406072903)

[8.1.34 <Artifact> IEPConformanceTargetType 20](#_Toc406072904)

[8.1.35 <Artifact> ModelPackageDescription 21](#_Toc406072905)

[8.1.36 <Artifact> OrganizationType 37](#_Toc406072906)

[8.1.37 <Artifact> PersonType 37](#_Toc406072907)

[8.1.38 <Artifact> QualifiedNamesType 38](#_Toc406072908)

[8.1.39 <Artifact> RelaxNGValidationType 38](#_Toc406072909)

[8.1.40 <Artifact> SchemaDocumentSet 38](#_Toc406072910)

[8.1.41 <Artifact> SchemaDocumentSetType 38](#_Toc406072911)

[8.1.42 <Artifact> SchematronValidationType 39](#_Toc406072912)

[8.1.43 <Artifact> TextRuleType 39](#_Toc406072913)

[8.1.44 <Artifact> ValidityConstraintType 39](#_Toc406072914)

[8.1.45 <Artifact> ValidityConstraintWithContextType 40](#_Toc406072915)

[8.1.46 <Artifact> ValidityContextType 40](#_Toc406072916)

[8.1.47 <Artifact> XMLSchemaType 40](#_Toc406072917)

[8.1.48 <Artifact> XPathType 40](#_Toc406072918)

[8.1.49 <Enumeration> ChangeCodeSimpleType 41](#_Toc406072919)

[8.1.50 <Enumeration> ModelPackageDescriptionClassCode 41](#_Toc406072920)

[8.1.51 <Enumeration> RelationshipCode 42](#_Toc406072921)

[8.2 Profile : NIEM\_Common\_Profile 43](#_Toc406072922)

[8.2.1 Overview 43](#_Toc406072923)

[8.2.2 <Stereotype> AdapterType 44](#_Toc406072924)

[8.2.3 <Stereotype> AssociationType 45](#_Toc406072925)

[8.2.4 <Stereotype> AugmentationType 46](#_Toc406072926)

[8.2.5 <Stereotype> Choice 48](#_Toc406072927)

[8.2.6 <Stereotype> Deprecated 49](#_Toc406072928)

[8.2.7 <Stereotype> Documentation 49](#_Toc406072929)

[8.2.8 <Stereotype> List 50](#_Toc406072930)

[8.2.9 <Stereotype> LocalTerm 52](#_Toc406072931)

[8.2.10 <Stereotype> LocalVocabulary 53](#_Toc406072932)

[8.2.11 <Stereotype> MetadataApplication 53](#_Toc406072933)

[8.2.12 <Stereotype> MetadataType 54](#_Toc406072934)

[8.2.13 <Stereotype> Namespace 55](#_Toc406072935)

[8.2.14 <Stereotype> NIEMType 66](#_Toc406072936)

[8.2.15 <Stereotype> ObjectType 66](#_Toc406072937)

[8.2.16 <Stereotype> PropertyHolder 72](#_Toc406072938)

[8.2.17 <Stereotype> References 73](#_Toc406072939)

[8.2.18 <Stereotype> Representation 74](#_Toc406072940)

[8.2.19 <Stereotype> Restriction 75](#_Toc406072941)

[8.2.20 <Stereotype> Union 76](#_Toc406072942)

[8.2.21 <Stereotype> UnionOf 79](#_Toc406072943)

[8.2.22 <Stereotype> ValueRestriction 80](#_Toc406072944)

[8.3 Profile : NIEM\_PIM\_Profile 84](#_Toc406072945)

[8.3.1 Overview 84](#_Toc406072946)

[8.3.2 <Stereotype> Augments 84](#_Toc406072947)

[8.3.3 <Stereotype> InformationModel 84](#_Toc406072948)

[8.3.4 <Stereotype> ReferenceName 110](#_Toc406072949)

[8.3.5 <Stereotype> RoleOf 111](#_Toc406072950)

[8.3.6 <Stereotype> RolePlayedBy 112](#_Toc406072951)

[8.3.7 <Stereotype> Subsets 112](#_Toc406072952)

[8.3.8 <Enumeration> DefaultPurposeCode 113](#_Toc406072953)

[8.4 Profile : NIEM\_PSM\_Profile 120](#_Toc406072954)

[8.4.1 Overview 120](#_Toc406072955)

[8.4.2 <Stereotype> XSDAnyProperty 120](#_Toc406072956)

[8.4.3 <Stereotype> XSDDeclaration 121](#_Toc406072957)

[8.4.4 <Stereotype> XSDProperty 121](#_Toc406072958)

[8.4.5 <Stereotype> XSDRepresentationRestriction 141](#_Toc406072959)

[8.4.6 <Stereotype> XSDSimpleContent 142](#_Toc406072960)

[8.4.7 <Enumeration> XSDProcessContentsCode 142](#_Toc406072961)

[8.4.8 <Enumeration> XSDPropertyKindCode 143](#_Toc406072962)

[8.4.9 <Enumeration> XSDWhiteSpaceCode 143](#_Toc406072963)

[8.5 Profile : NIEM\_UML\_Profile 144](#_Toc406072964)

[8.5.1 Overview 144](#_Toc406072965)

[9 NIEM-UML Transformation Reference 145](#_Toc406072966)

# Submission-related material

## Submission Introduction

The NIEM-UML-3 submission team is pleased to present an initial submission to the “UML Profile for NIEM 3” Request for Proposal gov/14-03-01.

The IPR mode for this submission is Non-Assert.

Clause 0 of this document contains information specific to the OMG submission process and is not part of the proposed specification. The proposed specification starts with Clause 1. All clauses are normative unless otherwise specified.

## Submission Team

### Submitters

* Model Driven Solutions, cory-c@modeldriven.com
* Office of the Program Manager for Information Sharing Environment (PM-ISE) [www.ise.gov](http://www.ise.gov)

### Government Stakeholders

* NIEM Program Management Office (PMO), and the NIEM Technical Architecture Committee (NTAC)
* Office of the Secretary of Defense

### Contributors

* Adaptive
* Escape Velocity
* Everware-CBDI
* Georgia Tech Research Institute (GTRI)
* Hidden Symmetry Ltd
* SEARCH
* TethersEnd Consulting

## Resolution of Requirements

### Mandatory requirements

|  |  |
| --- | --- |
| 6.5.1.1 Proposals shall specify a PIM profile constrained to limit the set of representable constructs to those allowed by the NIEM specifications. | The PIM profile is a modified version of the NIEM 1 PIM profile, altered to account for NIEM 3 changes. |
| 6.5.1.2 Proposals shall specify a MPD profile to specify the MPD metadata, content and technology choices required to parameterize the QVT transformations. | The MPD profile is a modified version of the NIEM 1 MPD profile, altered to account for NIEM 3 changes. |
| 6.5.1.3 Proposals shall specify a PSM profile providing stereotypes that enable NIEM technical modelers – or NIEM schema modelers – to model the technical aspect, i.e. XML schema specifics, of an MPD. | The PSM profile is a modified version of the NIEM 1 PSM profile, altered to account for NIEM 3 changes. |
| 6.5.1.4 Proposals shall specify QVT transformations from UML models using the PIM, MPD and PSM profiles to the set of artifacts required in a conformant MPD. | The QVT transformations are a modified versions of the NIEM 1 QVT transformations, altered to account for NIEM 3 changes. |
| 6.5.1.5 Proposals shall utilize the PIM, MPD and PSM profiles to model at least one existing NIEM IEPD and demonstrate the resulting transformation to a NIEM-conformant IEPD. | To be discussed in revised submission. NIEM 3 is not yet finalized. |
| 6.5.2.1 Proposals shall include UML models of the NIEM Version 3 reference schema namespaces [NIEM-RN] (or vocabularies) using the PIM; each UML model shall contain a package representing a NIEM namespace. | To be discussed in revised submission. NIEM 3 is not yet finalized. |
| 6.5.3.1 Proposals shall, wherever practical, reuse elements from the UML Profile for NIEM Version 1.0 PIM profile and provide for forward compatibility of models developed based on the profile. | The version 1.0 profile has been reused to the maximum extent possible. |
| 6.5.3.2 Proposals shall discuss the relationship between the UML Profile for NIEM, Version 1.0 (aligned to NIEM 2.1) and the UML Profile for NIEM Version 3 and the conversion process users would be expected to follow. | To be discussed in revised submission. |
| 6.5.3.3 Requirements outlined in sections 6.5.1 through 6.5.3 shall conform to normative NIEM specifications referenced in section 6.4. | The specification is extensively cross-referenced to the normative NIEM 3 specifications. |

### Non-mandatory features

|  |  |
| --- | --- |
| 6.6.1 Proposals may include a provisioning profile and mapping to other exchange formats such as RDF Schema (http://www.w3.org/standards/techs/rdf#w3c\_all). | May be addressed in revised submission. |
| 6.6.2 Proposals may specify a “reverse engineering” OMG-QVT mapping from a NIEM-conformant MPD to UML models that conform to the NIEM-UML profiles. | May be addressed in revised submission. |
| 6.6.3 Proposals may specify a transformation from the UML Profile for NIEM, Version 1.0 (aligned to NIEM 2.1) and the UML Profile for NIEM Version 3 conformant models. | May be addressed in revised submission. |
|  |  |

## Resolution of Discussion Issues

|  |
| --- |
| 6.7.1 Proposals shall discuss the relationship of NIEM-UML with other on-going and related NIEM standards, the existing NIEM specifications and the NIEM process. |

To be discussed in revised submission.

|  |
| --- |
| 6.7.2 Proposals shall discuss the relationship of NIEM-UML with other on-going exchange standards efforts, including but not limited to the Structured Threat Information Expression (STIX) and EDXL. |

To be discussed in revised submission.

|  |
| --- |
| 6.7.3 Proposals shall discuss their relationship with other relevant standards including but not limited to the Unified Profile for DoDAF/MODAF (UPDM) 2.1. |

To be discussed in revised submission.

# Preface

**OMG**

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies and academia. OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets. More information on the OMG is available at [*http://www.omg.org/*](http://www.omg.org/)*.*

**OMG Specifications**

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Specifications are available from this URL: [*http://www.omg.org/spec*](http://www.omg.org/spec)

Specifications are organized by the following categories:

**Business Modeling Specifications**

**Middleware Specifications**

* CORBA/IIOP
* Data Distribution Services
* Specialized CORBA IDL/Language Mapping Specifications

**Modeling and Metadata Specifications**

* UML, MOF, CWM, XMI
* UML Profile  Specifications

**Platform Independent Model (PIM) - Platform Specific Model (PSM) - Interface Specifications**

* CORBAServices
* CORBAFacilities
* OMG Domain Specifications
* CORBA Embedded Intelligence Specifications
* CORBA Security Specifications

All of OMG’s formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at: OMG Headquarters 140 Kendrick Street Building A, Suite 300 Needham, MA 02494 USA Tel: +1- 781-444-0404 Fax: +1-781-444-0320 Email: *pubs@omg.org* Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

**Typographical Conventions**

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

**Helvetica/Arial - 10 pt. Bold:** OMG Interface Definition Language (OMG IDL) and syntax elements.

**Courier - 10 pt. Bold:** Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

**Note –** Terms that appear in *italics* are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

**Issues**

The reader is encouraged to report any technical or editing issues/problems with this specification to *http://www.omg.org/ report\_issue.htm.*

# Scope

# Conformance

# Normative References

# Terms and Definitions

# Symbols

There are no symbols defined in this specification.

# Additional Information

# NIEM-UML Modeling Guide

# NIEM-UML Profile Reference

## Profile : Model\_Package\_Description\_Profile

### Overview

The Model Package Description Profile comprises stereotypes that are used to model NIEM MPDs. The diagram shows all the stereotypes defined in this profile.

### <Stereotype> ApplicationInfo

##### Description

An MPD artifact that is used by a software tool (e.g., import, export, input, output, etc.).

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> BusinessRulesArtifact

##### Description

An MPD artifact that contains business rules and constraints on exchange content.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> ChangeInformationType

##### Description

The «ChangeInformationType» stereotype applies to a Package that represents one or more detailed change entries. The «ChangeInformationType» is a nested UML::Package of «ChangeLogType». It contains descriptive information about one or more detailed change entries. The attributes defined for «ChangeInformationType» reflect the required changelog descriptive information for change entries. The change entries themselves, and their relationship with «ChangeInformationType» is an implementation detail not constrained by this specification.

##### Extends

UML::Package

##### Properties

###### ChangeCode : ChangeCodeSimpleType [0..\*]

An enumeration of change codes based on the type of change that is contained in the change log.

###### ChangeFullDescriptionText : String [0..1]

Descriptive text outlining the details of a specific change contained in a change log.

###### ChangeNCCTIssueNumber : Integer [0..\*]

Text outlining the NIEM Change Configuration Tool number associated to the specific change contained in the change log.

###### ChangeReasonText : String [0..1]

Descriptive text providing context to the reason a change noted in the change log was made.

###### ChangeSummaryText : String [0..1]

Text outlining a summary of a specific change contained in the change log.

### <Stereotype> ChangeLogType

##### Description

The ChangeLogType stereotype applies to a Package that represents the required MPD changelog artifact. The changelog artifact contains descriptive information about the changelog as a whole. The attributes defined for «ChangeLogType» reflect the required changelog descriptive information.

##### Extends

UML::Package

##### Properties

###### BaselineModelURL : String [1]

URL of baseline model the change log applies to.

###### ChangeLogApplicationInstructionsText : String [0..1]

Descriptive text representing change log applications instructions.

###### ChangeLogSubmitterName : String [0..1]

A name of the person, or organization submitting the change log.

###### ChangeLogSummaryText : String [0..1]

Descriptive text providing a summary of the change log.

### <Stereotype> ConformanceAssertion

##### Description

An MPD artifact that represents a declaration that a NIEM IEPD or EIEM is NIEM-conformant.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> ConformanceReport

##### Description

An MPD artifact either auto-generated by a NIEM-aware software tool or manually prepared that checks NIEM conformance and/or quality and renders a detailed report of results. This report may also be an auto-generated and manually prepared hybrid artifact.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> Documentation

##### Description

An MPD artifact that is a form of explanatory documentation.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> ExtensionSchemaDocument

##### Description

An MPD artifact that is a NIEM extension schema document.

##### Generalization

[XMLSchemaDocument](#_b02fa05d43cb5f5cace47c9181b17443)

### <Stereotype> ExternalSchemaDocument

##### Description

An MPD artifact that is a schema document external to NIEM.

##### Generalization

[XMLSchemaDocument](#_b02fa05d43cb5f5cace47c9181b17443)

### <Stereotype> File

##### Description

A generic electronic file artifact in an MPD; a file stored on a computer system.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> FileType

##### Description

A data type for an MPD file artifact.

##### Extends

UML::Usage

##### Properties

###### descriptionText : String [0..1]

A description of the file. Implemented as the value of the descriptionText attribute of the File element in the catalog instance.

###### externalURI : String [0..1]

An external URI for the file; indicates a same-as relationship to a copy of the file. Implemented as the value of the externalURI attribute of the File element in the catalog instance.

###### mimeMediaTypeText : String [0..1]

A classification for an MPD file artifact from the IANA MIME media classes: <http://www.iana.org/assignments/media-types>.

###### pathURI : String [1]

The relative path name to the file within the MPD directory structure. Implemented as the value of the pathURI attribute of the FileType type in the catalog instance.

### <Stereotype> IEPSampleXMLDocument

##### Description

An example MPD instance XML document or IEP artifact.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> ModelPackageDescriptionRelationship

##### Description

The ModelPackageDescriptionRelationship stereotype applies to a Dependency that represents a relationship between MPDs or between an MPD and another resource (such as a NIEM specification; as in the case of conforms-to). There are many ways one MPD may relate to another. This makes it difficult to specify a fixed set of values that could objectively define an exact relationship between a pair of MPDs. Therefore, the optional descriptionText attribute is provided to further explain the nature of any of the relationshipCode values available (version\_of, specializes, generalizes, deprecates, supersedes, adapts, conforms\_to, updates, derives\_from). In some cases, the value of relationshipCode may be generic enough to require a more detailed explanation in descriptionText (for example, if the value is "adapts").

##### Extends

UML::Dependency

##### Properties

###### descriptionText : String [0..1]

A more detailed or specific textual explanation of the relationship between the MPDs or between an MPD and a resource (such as a specification).

###### relationshipCode : RelationshipCode [1]

A classification or reason for the connectedness between the MPDs or between an MPD and a resource.

### <Stereotype> MPDChangeLog

##### Description

An MPD artifact that contains a record of the MPD changes.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> ReadMe

##### Description

An MPD read-me artifact.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> ReferenceSchemaDocument

##### Description

An MPD artifact that is a reference schema document (from a release, domain update, or core update).

##### Generalization

[XMLSchemaDocument](#_b02fa05d43cb5f5cace47c9181b17443)

### <Stereotype> RelaxNGSchema

##### Description

A RelaxNG schema.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> RequiredFile

##### Description

An MPD file artifact that another artifact depends on and should not be separated from.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> SchematronSchema

##### Description

A Schematron schema document.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> SubsetSchemaDocument

##### Description

An MPD artifact that is a subset schema document.

##### Generalization

[XMLSchemaDocument](#_b02fa05d43cb5f5cace47c9181b17443)

### <Stereotype> Wantlist

##### Description

An MPD artifact that represents a NIEM schema subset and is used as an import or export for the NIEM SSGT. See [Section 6.1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_6.1) of [NIEM-MPD].

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> XMLCatalog

##### Description

An MPD artifact that is an OASIS XML catalog.

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Stereotype> XMLSchemaDocument

##### Description

An MPD artifact that is an XML schema document (i.e., an XSD that is not necessarily a NIEM subset, extension, or reference schema).

##### Generalization

[FileType](#_1adfac21ca9bec3d93546a3277176cea)

### <Artifact> ArtifactOrArtifactSet

##### Description

A data concept for a file or file set in an MPD.

### <Artifact> ConformanceTargetType

##### Description

A data type for identifying and describing a conformance target.

##### Generalization

[ValidityConstraintType](#_9863680455d9ae6e5de2db10c1a2edbc)

##### Properties

###### conformanceTargetURI : String [1]

A URI for a conformance target.

### <Artifact> ConstraintSchemaDocumentSet

##### Description

An MPD artifact set of constraint schema documents and other supporting artifacts.

##### Generalization

[ArtifactOrArtifactSet](#_db7585560e9f04d9f4eaefd6a5b723fc)[SchemaDocumentSetType](#_03c988919690710762e6cd5069e6c488)

### <Artifact> ContactInformationType

##### Description

A data type for how to contact a person or an organization.

##### Properties

###### ContactEmailID : String [0..\*]

An electronic mailing address by which a person or organization may be contacted.

###### ContactEntity : EntityRepresentation [0..\*]

An entity that may be contacted by using the given contact information.

###### ContactMailingAddress : String [0..\*]

A postal address by which a person or organization may be contacted.

###### ContactResponder : PersonType [0..\*]

A third party person who answers a call and connects or directs the caller to the intended person.

###### ContactTelephoneNumber : String [0..\*]

A telephone number for a telecommunication device by which a person or organization may be contacted.

###### ContactWebsiteURI : String [0..\*]

A website address by which a person or organization may be contacted.

### <Artifact> DescribedType

##### Description

Common supertype for NIEM MPD Catalog types which have descriptionText.

##### Properties

###### descriptionText : String [0..1]

A description of the file. Implemented as the value of the descriptionText attribute of the File element in the catalog instance.

### <Artifact> EntityRepresentation

##### Description

A data concept for a person, organization, or thing capable of bearing legal rights and responsibilities.

##### Properties

###### name : String [0..\*]

A combination of names and/or titles by which an entity is known.

### <Artifact> EXIXMLSchemaType

##### Description

An XML Schema to be used for EXI serialization of an IEP Class.

##### Generalization

[ArtifactOrArtifactSet](#_db7585560e9f04d9f4eaefd6a5b723fc)[XMLSchemaType](#_5e48a7ce22c2dd0be5cc1ed17c6e3f38)

### <Artifact> FileSet

##### Description

A generic MPD artifact set; used to group artifacts that are not accounted for by other set classifiers.

##### Generalization

[ArtifactOrArtifactSet](#_db7585560e9f04d9f4eaefd6a5b723fc)[FileSetType](#_f9ab110d19d406069517dfa76824683e)

### <Artifact> FileSetType

##### Description

A data type for a set of MPD file artifacts.

##### Generalization

[DescribedType](#_decd4bb0486b2d193f0fe691f9f3e96f)

##### Properties

###### ArtifactOrArtifactSet : ArtifactOrArtifactSet [0..\*]

A data concept for a file or file set in an MPD.

###### externalURI : String [0..1]

An external URI for the file; indicates a same-as relationship to a copy of the file. Implemented as the value of the externalURI attribute of the File element in the catalog instance.

###### pathURI : String [1]

The relative path name to the file within the MPD directory structure. Implemented as the value of the pathURI attribute of the FileType type in the catalog instance.

### <Artifact> IEPConformanceTargetType

##### Description

A data type for a class or category of IEP, which has a set of validity constraints and a unique identifier.

##### Generalization

[DescribedType](#_decd4bb0486b2d193f0fe691f9f3e96f)

##### Properties

###### ArtifactOrArtifactSet : ArtifactOrArtifactSet [0..\*]

A data concept for a file or file set in an MPD.

###### ValidityConstraintWithContext : ValidityConstraintWithContextType [0..\*]

A data concept for a rule or instructions for validating an IEP candidate (XML document) using some context within that XML document.

##### Constraints

###### MPD3 [Rule 5-44] (IEPD). IEPD Has an IEP Sample for Each c:IEPConformanceTarget

[Rule 5-44](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-44), IEPD Has an IEP Sample for Each *c:IEPConformanceTarget*: [Section 5.6.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.6.3), IEP Sample Instance XML Documents

**[OCL] context** IEPConformanceTargetType **inv:**

self.oclAsType(InstanceSpecification).clientDependency->exists(d|d.stereotypedBy('IEPSampleXMLDocument'))

###### MPD3 [Rule 5-45] (IEP). Validating an IEP Sample XML Document

[Rule 5-45,](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-45) Validating an IEP Sample XML Document: [Section 5.6.3,](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.6.3) IEP Sample Instance XML Documents

[English]

The constraint is enforced during provisioning of the Sample XML Document.

### <Artifact> ModelPackageDescription

##### Description

A ModelPackageDescription Artifact represents a NIEM Model Package Description (MPD). Specifically, it represents the information in the NIEM-3 MPD catalog, which is defined for target namespace=http://reference.niem.gov/niem/resource/mpd/catalog/3.0/, in the context of a NIEM-3 subset of the niem-core schema.

An MPD is a logical set of electronic files aggregated and organized to fulfill a specific purpose in NIEM. Directory organization and packaging of an MPD should be designed around major themes in NIEM: reuse, sharing, interoperability, and efficiency. The inclusion of artifacts in an MPD is modeled using a Usage dependency from the InstanceSpecification representing the MPD to the model element representing the artifact (most commonly a Namespace Package).

The attributes of the ModelPackageDescription correspond to components of the MPDType within the MPD Catalog Schema.  The information model fragment corresponding to c:MPDInformation has been partially flattened from the schema containment structure into the ModelPackageDescription Attributes.

In addition to the largely isomorphic representation of the MPD Catalog as parts of the ModelPackageDescription, there are a few convenience mechanisms to simplify the UML Model.   The representation of the MPD relationship between MPDs is modeled as a «ModelPackageDescriptionRelationship» Dependency from the client ModelPackageDescription Artifact Instance to the related ModelPackageDescription Artifact Instance as supplier.

Instances of ModelPackageDescription may be the client of a UML Usage to some supplying NIEM concept, such as «InformationModel».  These UML Usages may be stereotyped with a sub-stereotype of «FileType».  When such a Usage is defined, values of some stereotype tags may be derived, such as the pathURI (based on UML Package Structure and/or NIEM packaging structure guidelines).  Note that most «FileType»s are implicit anyway, being derived by transitive closure of all «InformationModel»s referenced by used «InformationModel»s.

##### Properties

###### ArtifactOrArtifactSet : ArtifactOrArtifactSet [0..\*]

A data concept for a file or file set in an MPD.

###### AuthoritativeSource : EntityRepresentation [0..\*]

An official sponsoring or authoring organization responsible for an MPD.

###### CreationDate : String [1]

Date this MPD was published or created. Implemented as the value of the CreationDate element in the catalog instance.

###### descriptionText : String [0..1]

A description of the MPD. A statement that provides an explanation or additional detail.

Implemented as the value of the DescriptionText element of the MPDType type within the MPD Catalog document instance.

###### DomainText : String [0..\*]

A NIEM Domain applicable to, associated with, or that uses the MPD. Implemented as the value of the DomainText element in the catalog instance.

###### ExchangePartnerName : String [0..\*]

Name of an agency, organization, or entity that uses the MPD (in particular to share or exchange data). Implemented as the value of the ExchangePartnerName element in the catalog instance.

###### ExchangePatternText : String [0..\*]

A description of a transactional, design, or exchange pattern the MPD uses (generally, applicable to IEPDs only). Implemented as the value of the ExchangePatternText element in the catalog instance.

###### IEPConformanceTarget : IEPConformanceTargetType [0..\*]

A class or category of IEPs which has a set of validity constraints and a unique identifier. Every IEP is an instance of one or more IEP Conformance Targets.

###### KeywordText : String [0..\*]

A keyword associated with the MPD; a common alias, term, or phrase that would help to facilitate search and discovery of this MPD. Implemented as the value of the KeywordText element in the catalog instance.

###### LastRevsionDate : String [0..1]

Date the MPD was last revised. Implemented as the value of the LastRevisionDate element in the catalog instance.

###### mpdBaseURI : String [1]

The left hand substring of an MPD URI that does not include its mpdVersionID. The concatenation of mpdBaseURI and mpdVersionID becomes the value of the mpdURI attribute of the MPD element in the catalog instance. The last segment of mpdBaseURI becomes the value of the mpdName attribute of the MPD element in the catalog instance.

Note that the relationship between mpdBaseURI, mpdURI, and mpdName are more restrictive than the rules expressed in NIEM 3 MPD, but are consistent with guidelines recommended/implied by the NIEM 3 MPD.

###### mpdClassCode : ModelPackageDescriptionClassCode [1]

The classification code of the MPD. Maps to the value of the mpdClassURIList attribute of the MPDType within the catalog instance. This code designates the classification or kind of the MPD.

Note that NIEM-3 MPD explicitly defines only the iepd classification code, with the other classification codes implied but not formally defined with the MPD specification.

###### mpdVersionID : String [1]

Many published MPDs will be periodically revised and updated; therefore, versioning is required to clearly indicate that changes have occurred. See [Section 5.2.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.3) of [NIEM-MPD]. A version number is actually part of the unique identification for an MPD. All NIEM version numbers adhere to the regular expression: [0-9]+(\.[0-9]+)\*((alpha|beta|rc|rev)[0-9]+)?

Where:

1. "alpha" indicates early development
2. "beta" indicates late development; but changing or incomplete
3. "rc" indicates release candidate; complete but not approved as operational
4. "rev" indicates very minor revision that does not impact schema validation

Note that the value of mpdVersionID is concatenated with mpdBaseURI to form the mpdURI.  This convention is more restrictive then the NIEM 3 MPD rules, but is consistent with recommended/implied MPD naming conventions.

###### PurposeText : String [0..\*]

A description for the purpose, function, intended use of, or reason for the existence of the MPD. Implemented as the value of the PurposeText element in the catalog instance.

###### StatusText : String [0..1]

Description of the current state of development or usage of the MPD; may also project future plans for the MPD. Implemented as the value of the StatusText element in the catalog instance.

##### Constraints

###### MPD [Rule 3-09]

**[Rule 3-9]** A NIEM-conforming IEPD or EIEM MUST contain at least one schema that is either a NIEM reference schema or a subset derived from a NIEM reference schema.

[English]

// pseudo code for specifying constraint in terms of tag values of foreign stereotype instances  
  
(  
 (self.mpdClassCode=ModelPackageDescriptionClassCode::iepd)   
 or (self.mpdClassCode=ModelPackageDescriptionClassCode::eiem)  
) implies  
 self.oclAsType(InstanceSpecification).clientDependency->select(d| d.stereotypedBy('XMLSchemaDocument'))  
 ->exists(xmlDoc|  
 xmlDoc.stereotypedBy('SubsetSchemaDocument')  
 or xmlDoc.stereotypedBy('ReferenceSchemaDocument')  
 or xmlDoc.supplier->exists(s|s.isDirectlyOrHasIndirectlyASubsetOrReferenceSchema())  
 or xmlDoc.supplier.getStereotypeApplication('InformationModel').defaultPurpose.name  
 ->exists(purposeURI|  
 (purposeURI='subset')  
 or  
 (purposeURI='reference')  
 )   
 )

###### MPD [Rule 3-10]

**[Rule 3-10]** A NIEM IEPD MUST contain at least one valid sample XML instance (i.e., IEP) artifact for each exchange schema element that can be the root of a corresponding IEP.

[English]

This constraint is realized by PSM-MPD transformations.

###### MPD [Rule 4-03.1]

**[Rule 4-3.1]** A higher MPD version number within a version series does NOT imply compatibility between versions. Compatibility between or among MPD versions MUST be explicitly stated in documentation.

[English]

Satisfaction of this constraint requires comparative analysis between versions; cannot be expressed easily in OCL.

###### MPD [Rule 4-06]

**[Rule 4-6]** Each file artifact in an MPD MUST have a corresponding File element in the catalog for that MPD.

[English]

Constraints for catalog construction are resolved in PSM-MPD transformation.

###### MPD [Rule 4-07]

**[Rule 4-7]** Each file set artifact in an MPD MUST have a corresponding FileSet element in the catalog for that MPD. This FileSet element must identify each file artifact that is a member of that file set artifact.

[English]

Constraints for catalog construction are resolved in PSM-MPD transformations.

###### MPD [Rule 4-08]

**[Rule 4-8]** Each artifact identified in the catalog MUST be assigned an id in the format of an NCName (Non-Colonized Name) as defined by [W3-XML-Namespaces]. This is required for both File and FileSet artifacts.

[English]

All catalog constraints are resolved in PSM-MPD transformation.

###### MPD [Rule 4-10]

**[Rule 4-10]** NIEM namespaces MUST NOT be used as URIs for MPD artifacts.

[English]

Constraints on artifact URIs are resolved during PSM-MPD transformations.

###### MPD [Rule 4-11]

**[Rule 4-11]** Every MPD that is a reference schema set (i.e., NIEM releases, core updates, and domain updates) MUST contain an XML change log artifact that:

* Validates with the NIEM change log schemas (mpd-changelog.xsd and niem-model.xsd).  Note: These are the base filenames; the actual filenames also contain a version number. For example: mpd-changelog-1.0.xsd is the current version.
* Records changes to previous reference schemas that this MPD represents.
* Bears the file name "changelog.xml".
* Resides in the root directory of the MPD.

[English]

Constraints on changelog are resolved during PSM-MPD transformations.

###### MPD [Rule 4-12]

**[Rule 4-12]** Every MPD that is an IEPD or EIEM MUST contain a change log artifact that:

* Records changes to previous IEPD or EIEM schemas that this MPD represents.
* Begins with the substring "changelog".
* Resides in the root directory of the MPD.

[English]

Constraints on changelogs are resolved by PSM-MPD transformations.

###### MPD [Rule 4-13.1]

**[Rule 4-13.1]** If an IEPD or EIEM contains more than one change log artifact, then each change log artifact MUST:

* Have a file name that begins with the substring "changelog".
* Reside in the MPD root directory .

[English]

Constraints on changelogs are resolved by PSM-MPD transformations.

###### MPD [Rule 4-13]

**[Rule 4-13]** The initial version of an IEPD or EIEM MUST contain a change log artifact with at least one entry for its creation date.

[English]

Constraints on changelogs are resolved during PSM-MPD transformation.

###### MPD [Rule 6-3b]

**[Rule 6-3b]** Within an MPD, the <name> and <version> substrings in the file name MUST match exactly the values for attributes mpdName and mpdVersionID within its catalog.xml artifact.

[English]

Packaging constraints are resolved by PSM-MPD transformations.

###### MPD [Rule 6-3c]

**[Rule 6-3c]** Within an MPD, the substring in the file name MUST correctly correspond to the value for the attribute mpdClassCode within catalog.xml. Correct correspondence is:

|  |  |
| --- | --- |
| IF file name <class> = | THEN catalog.xml mpdClassCode = |
| rel | release |
| cu | core-update |
| du | domain-update |
| iepd | iepd |
| eiem | eiem |

[English]

Packaging constraints are resolved by PSM-MPD transformations.

###### MPD [Rule 6-7]

**[Rule 6-7]** A published IEPD MUST link (through its catalog) to any EIEM it is based on.

[English]

An EIEM is an MPD with a packageCode of EIEM. An EIEM is typically bundled as a reusable model library which can be referenced from IEPDs. This relationship betweenEIEM and IEPD is used by PSM-MPD transformations to construct the catalog entries in resolution of this constraint.

###### MPD [Rule 6-8]

**[Rule 6-8]** Within an MPD archive, if non-NIEM-conforming schemas from other standards are used and referenced within an MPD, then all xsd:import, xsd:include, and xsd:redefine constructs used within those schemas MUST be modified as needed to have a value for the schemaLocation attribute that is a relative path reference that resolves to the correct schema within the sub-tree.

[English]

The schemaLocation constraints are resolved during PSM-MPD transformation.

###### MPD3 [Rule 3-2] (MPD). MPD with MPD class of IEPD is an IEPD

[Rule 3-2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_3-2), MPD with MPD class of IEPD is an IEPD: [Section 3.2.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_3.2.2), IEPD Conformance Target

[English]

Rule is definitional

###### MPD3 [Rule 3-3] (IEPD). IEPD Conformance Target Identifier

[Rule 3-3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_3-3), IEPD Conformance Target Identifier: [Section 3.2.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_3.2.2), IEPD Conformance Target

[English]

Constraint realized during provisioning of MPD Catalog.

###### MPD3 [Rule 4-1] (Schema-subset). Fundamental NIEM Subset Rule

[Rule 4-1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_4-1), Fundamental NIEM Subset Rule: [Section 4.2.1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_4.2.1), Basic Subset Concepts

[English]

Rule is definitional.

###### MPD3 [Rule 5-10] (WF-MPD). MPD Version Number Syntax

[Rule 5-10](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-10), MPD Version Number Syntax: [Section 5.2.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.3), MPD Version Numbering Scheme (c:mpdVersionID)

**[OCL] context** ModelPackageDescription **inv:**

self.mpdVersionID.match('[0-9]+(\\.[0-9]+)\*((alpha|beta|rc|rev)[0-9]+)?')

###### MPD3 [Rule 5-11] (WF-MPD). MPD URI Is Absolute

[Rule 5-11](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-11), MPD URI Is Absolute: [Section 5.2.4.1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.1), MPD URI Scheme (c:mpdURI)

[English]

Expressing constraint in OCL is deferred.

###### MPD3 [Rule 5-12] (WF-MPD). MPD URI Supports Fragment

[Rule 5-12](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-12), MPD URI Supports Fragment: [Section 5.2.4.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.2), URI Scheme for MPD Artifacts (c:externalURI)

[English]

Constraint is definitional.

###### MPD3 [Rule 5-13] (WF-MPD). MPD URI Has No Fragment [Rule 5-13] (WF-MPD) (Constraint)

[Rule 5-13](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-13), MPD URI Has No Fragment: [Section 5.2.4.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.2), URI Scheme for MPD Artifacts (c:externalURI)

[English]

Constraint is definitional.

###### MPD3 [Rule 5-14] (WF-MPD). MPD Artifact URI Syntax

[Rule 5-14](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-14), MPD Artifact URI Syntax: [Section 5.2.4.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.2), URI Scheme for MPD Artifacts (c:externalURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-15] (WF-MPD). c:pathURI Resolves to a Resource

[Rule 5-15](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-15), c:pathURI Resolves to a Resource: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog. c:pathURI is either set implicitly to a location, or if a value is provided, the resource is moved to specified location.

###### MPD3 [Rule 5-16] (WF-MPD). c:pathURI for c:XMLCatalog

[Rule 5-16](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-16), c:pathURI for c:XMLCatalog: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-17] (WF-MPD). c:pathURI for c:MPDChangeLog

[Rule 5-17](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-17), c:pathURI for c:MPDChangeLog: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog. Provisioning set values of c:pathURI based on relative location of changelog package.

###### MPD3 [Rule 5-18] (WF-MPD). c:pathURI for c:ReadMe

[Rule 5-18](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-18), c:pathURI for c:ReadMe: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-19] (WF-MPD). c:pathURI for c:IEPSampleXMLDocument

[Rule 5-19](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-19), c:pathURI for c:IEPSampleXMLDocument: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-1] (WF-MPD). MPD Has an mpd-catalog.xml in its Root Directory

[Rule 5-1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-1), MPD Has an mpd-catalog.xml in its Root Directory: [Section 5.1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.1), NIEM MPD Catalog

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-20] (WF-MPD). c:pathURI for c:BusinessRulesArtifact

[Rule 5-20](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-20), c:pathURI for c:BusinessRulesArtifact: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-21] (WF-MPD). c:pathURI for c:XMLSchemaDocument

[Rule 5-21](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-21), c:pathURI for c:XMLSchemaDocument: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-22] (WF-MPD). c:pathURI for c:ExternalSchemaDocument

[Rule 5-22](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-22), c:pathURI for c:ExternalSchemaDocument: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-23] (WF-MPD). c:pathURI for c:ReferenceSchemaDocument

[Rule 5-23](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-23), c:pathURI for c:ReferenceSchemaDocument: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-24] (WF-MPD). c:pathURI for c:ExtensionSchemaDocument

[Rule 5-24](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-24), c:pathURI for c:ExtensionSchemaDocument: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-25] (WF-MPD). c:pathURI for c:SubsetSchemaDocument

[Rule 5-25](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-25), c:pathURI for c:SubsetSchemaDocument: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-26] (WF-MPD). c:pathURI for c:Wantlist

[Rule 5-26](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-26), c:pathURI for c:Wantlist: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-27] (WF-MPD). c:pathURI for c:SchematronSchema

[Rule 5-27](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-27), c:pathURI for c:SchematronSchema: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-28] (WF-MPD). c:pathURI for c:RelaxNGSchema

[Rule 5-28](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-28), c:pathURI for c:RelaxNGSchema: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-29] (WF-MPD). c:pathURI for c:SchemaDocumentSet

[Rule 5-29](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-29), c:pathURI for c:SchemaDocumentSet: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-2] (MPD-catalog). MPD Catalog Document Valid to mpd-catalog-3.0.xsd

[Rule 5-2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-2), MPD Catalog Document Valid to mpd-catalog-3.0.xsd: [Section 5.1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.1), NIEM MPD Catalog

[English]

Concept and constraint not implemented in NIEM-UML.

###### MPD3 [Rule 5-30] (WF-MPD). c:pathURI for c:ConstraintSchemaDocumentSet

[Rule 5-30](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-30), c:pathURI for c:ConstraintSchemaDocumentSet: [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-31] (WF-MPD).

[Rule 5-31](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-31): [Section 5.2.4.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.3), URI Scheme for Local MPD Artifacts (c:pathURI)

[English]

Constraint is definitional.

###### MPD3 [Rule 5-32] (WF-MPD). Resolve MPD URI with Fragment

[Rule 5-32](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-32), Resolve MPD URI with Fragment: [Section 5.2.4.5](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.5), Resolving an MPD URI with a Fragment

[English]

Constraint is definitional.

###### MPD3 [Rule 5-33] (XML-catalog). XML Catalog uri Value Resolves to Resource

[Rule 5-33](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-33), XML Catalog uri Value Resolves to Resource: [Section 5.2.4.7](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.7), XML Catalog URI

[English]

Constraint is realized during provisioning of XML Catalog.

###### MPD3 [Rule 5-34] (XML-catalog). XML Catalog uri Value Resolves to Resource with Correct Target Namespace

[Rule 5-34](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-34), XML Catalog uri Value Resolves to Resource with Correct Target Namespace: [Section 5.2.4.7](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.7), XML Catalog URI

[English]

Constraint is realized during provisioning of MPD Catalog.

###### MPD3 [Rule 5-35] (IEPD). IEPD Has a Change Log

[Rule 5-35](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-35), IEPD Has a Change Log: [Section 5.3.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.3.2), Change Log for IEPDs

**[OCL] context** ModelPackageDescription **inv:**

self.oclAsType(InstanceSpecification).clientDependency->select(d|d.stereotypedBy('MPDChangeLog'))->notEmpty()

###### MPD3 [Rule 5-36] (WF-MPD). Readme Describes Purpose, Scope, Business Value, etc.

[Rule 5-36](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-36), Readme Describes Purpose, Scope, Business Value, etc.: [Section 5.4](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.4), ReadMe Artifact

[English]

The readme artifact provisioned includes purpose, scope, business value, exchange information, typical senders/receivers, interactions, and references to other documentation. This information is obtained from certain modeling conventions in the model.

###### MPD3 [Rule 5-37] (IEPD). IEPD Has a ReadMe Artifact

[Rule 5-37](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-37), IEPD Has a ReadMe Artifact: [Section 5.4](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.4), ReadMe Artifact

**[OCL] context** ModelPackageDescription **inv:**

self.oclAsType(InstanceSpecification).clientDependency->select(d|d.stereotypedBy('ReadMe'))->notEmpty()

###### MPD3 [Rule 5-38] (MPD-catalog). Conformance Target Identifier

[Rule 5-38](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-38), Conformance Target Identifier: [Section 5.6](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.6), Defining Information Exchange Packages

[English]

Rule enforced during provisioning. The target identifier is synthesized.

###### MPD3 [Rule 5-39] (MPD-catalog). IEP Conformance Target Has a structures:id

[Rule 5-39](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-39), IEP Conformance Target Has a structures:id: [Section 5.6](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.6), Defining Information Exchange Packages

[English]

Rule enforced during provisioning. The target structure:id attribute is synthesized.

###### MPD3 [Rule 5-3] (MPD-catalog). MPD Catalog Extension XML Catalog Document in Root Directory

[Rule 5-3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-3), MPD Catalog Extension XML Catalog Document in Root Directory: [Section 5.1.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.1.2), Extending an MPD Catalog

[English]

Concept and constraint not implemented in NIEM-UML.

###### MPD3 [Rule 5-40] (IEPD). IEPD Declares One or More IEP Conformance Targets

[Rule 5-40](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-40), IEPD Declares One or More IEP Conformance Targets: [Section 5.6](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.6), Defining Information Exchange Packages

[English]

Rule enforced during provisioning. Multiple c:IEPConformanceTarget elements are produced based on MPD specification.

###### MPD3 [Rule 5-41] (MPD-catalog).

[Rule 5-41](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-41): [Section 5.6.2.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.6.2.3), c:ValidityContext

[English]

Rule is definitional.

###### MPD3 [Rule 5-42] (IEP). Identifying the Document Element of an IEP

[Rule 5-42](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-42), Identifying the Document Element of an IEP: [Section 5.6.2.4](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.6.2.4), c:HasDocumentElement

In UML, the QualifiedNamesType represents the c:HasDocumentElement....

[English]

Probably some model adjustment here to make the concept of documentElement more clear.

###### MPD3 [Rule 5-43] (IEP). Validating an XPath Expression

[Rule 5-43](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-43), Validating an XPath Expression: [Section 5.6.2.5](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.6.2.5), c:ValidToXPath

[English]

Probably some model adjustment here to make the concept of validToXPath more clear.

###### MPD3 [Rule 5-46] (IEPD). IEPD Has Conformance Assertion

[Rule 5-46](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-46), IEPD Has Conformance Assertion: [Section 5.7](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.7), Conformance Assertion

[English]

self.clientDependency->select(d|d.oclIsKindOf(ConformanceAssertion))->notEmpty()

###### MPD3 [Rule 5-4] (MPD-catalog). MPD Catalog Extension XML Catalog Document Name Is mpd-catalog-extension-xml-catalog.xml

[Rule 5-4](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-4), MPD Catalog Extension XML Catalog Document Name Is mpd-catalog-extension-xml-catalog.xml: [Section 5.1.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.1.2), Extending an MPD Catalog

[English]

Concept and constraint not implemented in NIEM-UML.

###### MPD3 [Rule 5-5] (MPD-catalog). MPD Catalog Extension XML Catalog Document Resolves Namespaces to URIs

[Rule 5-5](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-5), MPD Catalog Extension XML Catalog Document Resolves Namespaces to URIs: [Section 5.1.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.1.2), Extending an MPD Catalog

[English]

Concept and constraint not implemented in NIEM-UML.

###### MPD3 [Rule 5-6] (MPD-catalog). MPD Catalog Extension Schema Document Conforms to NDR Extension Rules

[Rule 5-6](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-6), MPD Catalog Extension Schema Document Conforms to NDR Extension Rules: [Section 5.1.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.1.2), Extending an MPD Catalog

[English]

Concept and constraint not implemented in NIEM-UML.

###### MPD3 [Rule 5-7] (MPD-catalog). MPD Catalog Schema and Its Extensions Conform to NDR Schema Set Rules

[Rule 5-7](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-7), MPD Catalog Schema and Its Extensions Conform to NDR Schema Set Rules: [Section 5.1.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.1.2), Extending an MPD Catalog

[English]

Concept and constraint not implemented in NIEM-UML.

###### MPD3 [Rule 5-8] (MPD-catalog). MPD Schema Document Extension Support Schemas Are Supersets of Spec Subsets

[Rule 5-8](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-8), MPD Schema Document Extension Support Schemas Are Supersets of Spec Subsets: [Section 5.1.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.1.2), Extending an MPD Catalog

[English]

Concept and constraint not implemented in NIEM-UML.

###### MPD3 [Rule 5-9] (WF-MPD). MPD Class Determined by Conformance Target Identifier in c:mpdClassURIList

[Rule 5-9](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_5-9), MPD Class Determined by Conformance Target Identifier in c:mpdClassURIList: [Section 5.2.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.2), MPD Class (c:mpdClassURIList)

[English]

Constraint realized during provisioning of MPD Catalog.

###### MPD3 [Rule 6-1] (WF-MPD). Wantlist Location

[Rule 6-1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_6-1), Wantlist Location: [Section 6.1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_6.1), NIEM Wantlist

**[OCL] context** ModelPackageDescription **inv:**

self.oclAsType(InstanceSpecification).clientDependency->select(d|d.stereotypedBy('WantList'))->notEmpty()

###### MPD3 [Rule 7-1] (WF-MPD). MPD Is a ZIP File

[Rule 7-1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_7-1), MPD Is a ZIP File: [Section 7](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_7), Organization, Packaging, and Other Criteria

[English]

Rule enforcement is provided by provisioning, which produces the zip file.

###### MPD3 [Rule 7-2] (WF-MPD). XSD and XML Documents Conform to Applicable NDR Conformance Targets

[Rule 7-2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_7-2), XSD and XML Documents Conform to Applicable NDR Conformance Targets: [Section 7](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_7), Organization, Packaging, and Other Criteria

[English]

Rule enforcement is provided by provisioning and NDR/MPD rules expressed in OCL and applied to the NIEM-UML model.

###### MPD3 [Rule 7-3] (WF-MPD). MPD Archive Uncompresses to a Single Root Directory

[Rule 7-3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_7-3), MPD Archive Uncompresses to a Single Root Directory: [Section 7](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_7), Organization, Packaging, and Other Criteria

[English]

Rule enforcement is provided by provisioning.

###### MPD3 [Rule 7-5] (IEPD). IEPD File Name Syntax

[Rule 7-5](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_7-5), IEPD File Name Syntax: [Section 7.2](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_7.2), IEPD File Name Syntax

[English]

Packaging constraints are resolved by PSM-MPD transformations.

###### MPD3 [Rule 7-6] (WF-MPD). MPD Reference to Resource Uses Common URI Scheme

[Rule 7-6](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_7-6), MPD Reference to Resource Uses Common URI Scheme: [Section 7.3](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_7.3), Artifact Links to Other Resources

[English]

Constraints on URIs are partially satisfied by specific URI Constraints expressed elsewhere in the NDR and MPD. For URI references embedded elsewhere in the model, it would be difficult to express the constraint in OCL. This constraint must be manually resolved by the modeler.

###### MPD3 [Rule 7-7] (WF-MPD). IEPD Completeness

[Rule 7-7](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_7-7), IEPD Completeness: [Section 7.4](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_7.4), IEPD Completeness

[English]

This constraint is resolved by PSM-MPD transformations.

###### MPD3 [Rule 7-8] (WF-MPD). MPD External Schema Documents Are Local Resources

[Rule 7-8](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_7-8), MPD External Schema Documents Are Local Resources: [Section 7.4](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_7.4), IEPD Completeness

[English]

This constraint is resolved by the NIEM-UML Model, which requires all InformationModels to be defined, and PSM-MPD transformations which enforce schemaLocations to be local.

###### MPD3 [Rule 7-9] (WF-MPD). Key MPD Resources Are Local Resources

[Rule 7-9](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_7-9), Key MPD Resources Are Local Resources: [Section 7.4](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_7.4), IEPD Completeness

[English]

This constraint is resolved by provisioning; all generated artifacts are local to the MPD and all references between them are relative.

### <Artifact> OrganizationType

##### Description

A data type for a body of people organized for a particular purpose.

##### Generalization

[EntityRepresentation](#_b5992f81868b11b29878271a69332828)

##### Properties

###### OrganizationPrimaryContactInformation : ContactInformationType [0..\*]

A preferred means of contacting an organization.

### <Artifact> PersonType

##### Description

Represents an AuthoritativeSource for the MPD corresponding to a niem-core:PersonType.   In this case, an InstanceSpecification of PersonType is mapped to the MPD Catalog element c:MPDInformationType/c:AuthoritativeSource/niem-core:EntityPerson (whose type is niem-core:PersonType).

An InstanceSpecification of PersonType may also represent a ContactEntity within a ContactInformationType.  In this case, the instance of PersonType is mapped to the MPD Catalog element .../niem-core:EntityOrganization/niem-core:OrganizationPrimaryContactInformation/niem-core:ContactEntity/niem-core:EntityPerson (whose type is niem-core:PersonType).

An InstanceSpecification of PersonType may also represent a ContactResponder within a ContactInformationType.  In this case, the instance of PersonType is mapped to the MPD Catalog element .../niem-core:EntityOrganization/niem-core:OrganizationPrimaryContactInformation/niem-core:ContactResponder/niem-core:EntityPerson (whose type is niem-core:PersonType).

##### Generalization

[EntityRepresentation](#_b5992f81868b11b29878271a69332828)

### <Artifact> QualifiedNamesType

##### Description

A data type for a set of qualified names.

##### Generalization

[ValidityConstraintWithContextType](#_dc6a1ac72032832f049a1273df5c4b46)

##### Properties

###### qualifiedNameList : String [0..\*]

A list of qualified names.

### <Artifact> RelaxNGValidationType

##### Description

A data type for a RelaxNG validation constraint, indicating a RelaxNG schema document against which an artifact may be validated, as well as a description of the validation roots for assessment of validity.

##### Generalization

[ValidityConstraintType](#_9863680455d9ae6e5de2db10c1a2edbc)

### <Artifact> SchemaDocumentSet

##### Description

An MPD artifact set that may include subset schema documents, extension and external schema documents, and other supporting artifacts.

##### Generalization

[ArtifactOrArtifactSet](#_db7585560e9f04d9f4eaefd6a5b723fc)[SchemaDocumentSetType](#_03c988919690710762e6cd5069e6c488)

### <Artifact> SchemaDocumentSetType

##### Description

A data type for an MPD artifact set that may include subset schema documents, extension schema documents, and external schema documents or constraint schema documents.

##### Generalization

[FileSetType](#_f9ab110d19d406069517dfa76824683e)

##### Constraints

###### MPD3 [Rule 7-4] (MPD-catalog). Constraint on Elements of Type c:SchemaDocumentSetType

[Rule 7-4](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#rule_7-4), Constraint on Elements of Type c:SchemaDocumentSetType: [Section 7.1.1](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_7.1.1), Constraint on Elements of Type c:SchemaDocumentSetType

**[OCL] context** SchemaDocumentSetType **inv:**

self.oclAsType(InstanceSpecification).clientDependency->select(d|d.stereotypedBy('XMLSchemaDocument')or d.stereotypedBy('XMLCatalog'))->notEmpty()

### <Artifact> SchematronValidationType

##### Description

A data type for a Schematron validation constraint, indicating a Schematron schema document against which an artifact may be validated as well as a description of the validation roots for assessment of validity.

##### Generalization

[ValidityConstraintType](#_9863680455d9ae6e5de2db10c1a2edbc)

### <Artifact> TextRuleType

##### Description

A data type for a rule drafted in a human language.

##### Generalization

[ValidityConstraintType](#_9863680455d9ae6e5de2db10c1a2edbc)

##### Properties

###### RuleText : String [1]

A rule written in a human language.

### <Artifact> ValidityConstraintType

##### Description

A data concept for a rule or instructions for validating an IEP candidate.

##### Generalization

[ValidityConstraintWithContextType](#_dc6a1ac72032832f049a1273df5c4b46)

### <Artifact> ValidityConstraintWithContextType

##### Description

A data concept for a rule or instructions for validating an IEP candidate (XML document) using some context within that XML document.

##### Generalization

[DescribedType](#_decd4bb0486b2d193f0fe691f9f3e96f)

### <Artifact> ValidityContextType

##### Description

A data type for a rule or instructions for validating an IEP candidate within context defined by an XPath expression.

##### Generalization

[ValidityConstraintWithContextType](#_dc6a1ac72032832f049a1273df5c4b46)

##### Properties

###### ValidityConstraint : ValidityConstraintType [1..\*]

A data concept for a rule or instructions for validating an IEP candidate.

###### xPathText : String [1]

An XPath expression.

### <Artifact> XMLSchemaType

##### Description

A data type for a validity constraint that indicating an XML Schema against which an artifact may be validated, or which can be used for other purposes. c:XMLSchemaDocument identifies the root or starting XML schema document.

##### Generalization

[ValidityConstraintType](#_9863680455d9ae6e5de2db10c1a2edbc)

### <Artifact> XPathType

##### Description

A data type for an XPath expression.

##### Generalization

[ValidityConstraintType](#_9863680455d9ae6e5de2db10c1a2edbc)

##### Properties

###### xPathText : String [1]

An XPath expression.

### <Enumeration> ChangeCodeSimpleType

##### Description

Purpose of change.

##### Literals

###### new\_requirement

###### bug\_fix

###### refactoring

###### harmonization

###### general\_improvement

### <Enumeration> ModelPackageDescriptionClassCode

##### Description

A specified classification (type or kind) of the MPD.

The MPD specification applies to all NIEM model package descriptions (MPDs).   Currently, MPDs include the following:

* A NIEM information exchange package documentation (IEPD) that defines a NIEM data exchange.
* A NIEM release (including a major, minor, or micro release) [as defined in the NIEM High-Level Version Architecture 1.0].
* A NIEM domain update [as described in NIEM Domain Update Specification 1.0]. (Note these are NOT the same as a NIEM domain schema document that is a part of a NIEM release).
* A NIEM core update to a NIEM release.
* A NIEM Enterprise Information Exchange Model (EIEM) on which one or more IEPDs can be based.

Of these kinds of MPDs, the only kind which is formally specified in NIEM-3 is an IEPD.  The NIEM-3 UML Models all kinds of MPD, and the kind is defined as the EnumerationLiterals of this ModelPackageDescriptionClassCode Enumeration.

The kind of MPD is reflected in the MPD Catalog c:mpdClassURIList attribute.  That attribute will be provisioned with the appropriate list of URIs based on the value of this ModelPackageDescriptionClassCode Enumeration.

##### Literals

###### eiem

An Enterprise Information Exchange Model (EIEM) is an MPD that incorporates BIECs that meet enterprise business needs for exchanging data using NIEM [NIEM-BIEC]. An EIEM is an adaptation of NIEM schemas, tailored and constrained for and by an enterprise. An EIEM will contain the following schemas that are commonly used or expected to be used by the authoring enterprise: one standard NIEM schema subset and one or more NIEM extension schemas that extend existing NIEM data components or establish new data components.

###### iepd

NIEM Information Exchange Package Documentation (IEPD) is an MPD that defines a recurring XML data exchange. An NIEM IEPD is a set of valid XML schemas that may include portions of NIEM Core schemas, portions of NIEM Domain schemas, enterprise-specific or IEPD-specific extension schemas, and at least one exchange schema that defines a document element (as defined in [W3-XML-InfoSet]). The schemas contained in an IEPD work together to define a class of XML instances that consistently encapsulate data for information exchanges. Each XML instance in this class validates against the set of XML schemas contained within the IEPD.

###### core\_update

When necessary, the NIEM PMO can publish a core update. This is essentially identical to a domain update in terms of structure and use, with two important exceptions. First, a core update records changes that apply to a particular NIEM core version or another core update. This also means it is applicable to all NIEM releases using that same core version. Second, a core update is never published to replace a NIEM core. It is intended to add new schemas, new data components, new code values, etc. to a core without waiting for the next major release. In some cases, minor modifications to existing data components are possible.

###### release

A NIEM release is an MPD containing a full set of harmonized reference schemas that coherently define all content within a single version of NIEM. NIEM releases include major, minor, and micro releases (as defined in the NIEM High Level Version Architecture (HLVA)).

###### domain\_update

A domain update is an MPD containing reference schemas that represent changes to NIEM domains. The [NIEM-HLVA] defines a domain update as both a process and a NIEM product. Through use and analysis of NIEM releases and published content, domain users will identify issues and new data requirements for the domain and sometimes Core. NIEM domains use these issues as the basis for incremental improvements, extensions, and proposed changes to future NIEM releases. Both the process and product of the process are referred to as domain update.

### <Enumeration> RelationshipCode

##### Description

The possible reasons for the connectedness between the MPDs or between an MPD and a resource. This enumeration defines the possible values for the relationshipCode attribute of the ModelPackageDescriptionRelationship stereotype. Reference [Section 5.2.4.4](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#section_5.2.4.4) and [Appendix A](http://reference.niem.gov/niem/specification/model-package-description/3.0/model-package-description-3.0.html#appendix_A) of [NIEM MPD].

##### Literals

###### updates

A relationshipCode value for indicating that this MPD is an incremental update to the referenced resource. Used by a core or domain update to identify the domain schema in a NIEM release being incrementally updated (not replaced).

###### conforms\_to

A relationshipCode value for indicating that this MPD conforms to the referenced specification or standard.

###### version\_of

A relationshipCode value for indicating that this MPD is a different version of the referenced MPD. This code value is only needed in cases where significant name changes might obscure the relationship to the previous version. For example, NIEM Justice 4.1 is a version of GJXDM 3.0.3.

###### specializes

A relationshipCode value for indicating that this MPD is a specialization of the referenced MPD. This value is the inverse of generalizes.

###### generalizes

A relationshipCode value for indicating that this MPD is a generalization of the referenced MPD. This value is the inverse of specializes.

###### supersedes

A relationshipCode value for indicating that this MPD replaces the referenced MPD.

###### deprecates

A relationshipCode value for indicating that content in this MPD is preferred over content in the referenced MPD; and at some time in the future will supersede the referenced MPD.

###### adapts

A relationshipCode value for indicating that this MPD is an adaptation of the referenced MPD.

###### derives\_from

A relationshipCode value for indicating that this MPD has been derived from another; used to indicate an IEPD is derived from an EIEM (may have other uses as well).

## Profile : NIEM\_Common\_Profile

### Overview

The NIEM Common Profile comprises stereotypes that are used in both the NIEM PIM Profile and the NIEM PSM Profile. In addition, the UML metamodel subset covered by the NIEM Common Profile also includes the metaclasses PrimitiveType, Enumeration, EnumerationLiteral, Property and Generalization, even though they are not specifically extended by any stereotypes in the profile.

### <Stereotype> AdapterType

##### Description

An AdapterType is a NIEMType Class that represents a NIEM adapter type. A NIEM adapter type is a NIEM object type that adapts external components for use within NIEM. External components are not NIEM-conforming (e.g., data components from other standards, e.g. GML, ISO, etc.). An adapter type creates a new class of object that embodies a single concept composed of external components. AdapterType is implemented in XML Schema as a complex type definition with complex content. See [NIEM NDR] [Section 10.2.3.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.2), *External adapter types*.

##### Generalization

[NIEMType](#_cddcf0aa38f9fb92183a65a83b2b548f)

##### Constraints

###### NDR3 [Rule 10-11] (REF,EXT). External adapter type not a base type

[Rule 10-11](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-11), External adapter type not a base type (REF, EXT): [Section 10.2.3.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.2), External adapter types

**[OCL] context** AdapterType **inv:**

self.base\_Class.\_directedRelationshipOfTarget->select(t|t.oclIsKindOf(Generalization) or t.stereotypedBy('Restriction'))->size()=0

###### NDR3 [Rule 10-12] (SET). External adapter type not a base type

[Rule 10-12](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-12), External adapter type not a base type (SET): [Section 10.2.3.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.2), External adapter types

**[OCL] context** AdapterType **inv:**

self.base\_Class.\_directedRelationshipOfTarget->select(t|t.oclIsKindOf(Generalization) or t.stereotypedBy('Restriction'))->size()=0

###### NDR3 [Rule 10-69] (REF). External adapter type indicator annotates complex type

[Rule 10-69](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-69), External adapter type indicator annotates complex type (REF): [Section 10.9.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1), The NIEM appinfo namespace

[English]

This constraint realized by provisioning:  
A Class stereotyped as AdapterType will result in production of appinfo:externalAdapterTypeIndicator attribute on the xs:complexType representing the AdapterType.

###### NDR3 [Rule 10-8] (REF,EXT). External adapter type has indicator

[Rule 10-8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-8), External adapter type has indicator (REF, EXT): [Section 10.2.3.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.2), External adapter types.

[English]

The constraint is resolved during provisioning:  
An AdapterType and only an AdapterType has the appinfo:externalAdapterTypeIndicator set to a value of true.

###### NDR3 [Rule 10-9] (REF,EXT). Structure of external adapter type definition follows pattern

[Rule 10-9](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-9), Structure of external adapter type definition follows pattern (REF, EXT): [Section 10.2.3.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.2), External adapter types.

**[OCL] context** AdapterType **inv:**

self.base\_Class.general->isEmpty()  
and  
self.base\_Class.clientDependency->select(d|d.stereotypedBy('Restriction'))->isEmpty()

### <Stereotype> AssociationType

##### Description

AssociationType is a NIEMType class that represents a NIEM association type. A NIEM association type establishes a relationship between objects, along with the properties of that relationship. A NIEM association is an instance of an association type. Associations are used when a simple NIEM property is insufficient to model the relationship clearly and when properties of a UML Association or AssociationClass may not necessarily be sufficient to reflect the variability of a NIEM association. Consequently, the AssociationType Stereotype is applied to a UML Class. Since an AssociationClass is also a Class, the AssociationType Stereotype may be applied to a UML AssociationClass where appropriate.  Note that a UML AssociationClass specializing another AssociationClass must have the same number of ends as the other AssociationClass and must have at least two ends. This UML constraint prevents the usage of AssociationClass to model abstract NIEM association types that are intended to be extended by subtypes with additional ends. A UML AssociationClass can specialize an abstract UML Class. AssociationType is implemented in XML Schema as a complex type definition with complex content. See [NIEM-NDR] [Section 10.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.3.1), *Association types*.

##### Generalization

[NIEMType](#_cddcf0aa38f9fb92183a65a83b2b548f)

##### Constraints

###### NDR3 [Rule 10-19] (REF,EXT). Association types is derived from association type

[Rule 10-19](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-19), Association types is derived from association type (REF, EXT): [Section 10.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.3.1), Association types

**[OCL] context** AssociationType **inv:**

(  
 (  
 self.stereotypedBy('AssociationType') or self.oclIsKindOf(AssociationClass)  
 )  
 implies   
 (  
 self.name.endsWith('AssociationType')  
 and  
 self.general  
 ->union(self.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
 ->forAll(c|c.stereotypedBy('AssociationType') or c.oclIsKindOf(AssociationClass))   
 )  
)  
and  
(  
 self.name.endsWith('AssociationType')  
 implies  
 self.stereotypedBy('AssociationType') or self.oclIsKindOf(AssociationClass)  
)

### <Stereotype> AugmentationType

##### Description

AugmentationType is a NIEMType Class that represents a NIEM augmentation type. A NIEM augmentation type is a complex type that provides a reusable block of data that may be added to object types or association types. An augmentation of an object type is a block of additional data that is an instance of an augmentation type, added to an object type to carry additional data beyond that of the original object definition. The applicability of an augmentation may be restricted using an Augments Generalization.  AugmentationType is implemented in XML Schema as a complex type definition with complex content. See [NIEM-NDR] [Section 10.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4), *Augmentations*.

##### Generalization

[NIEMType](#_cddcf0aa38f9fb92183a65a83b2b548f)

##### Constraints

###### NDR [Rule 7-14]

**[Rule 7-14] (REF, EXT)** A component definition SHALL begin with a standard opening phrase that depends on the class of the component per Table 7-1 in NDR: Standard Opening Phrases:

**Rationale** A standard opening phrase based on component class helps to ensure consistent definitions that appropriate for the type of component item being defined. These opening phrases also provide a cue that facilitates recognition of the particular kind of component.

**[OCL] context** AugmentationType **inv:**

self.base\_Class.ownedComment->exists(documentation|  
 documentation.stereotypedBy('Documentation') and documentation.\_'body'.startsWith('A data type that supplements ')  
 )

###### NDR [Rule 7-47]

**NDR [Rule 7-47] (REF, EXT)** Essentially states that the general of the <Augmentation>, if it exists, must be an <Augmentation>.

Due to schema and other NDR rules, this also requires

* a maximum of one general for the given <Augmentation>
* any subtypes of the given <Augmentation> must also be <Augmentation>

**[OCL] context** AugmentationType **inv:**

(self.base\_Class.general->size()<=1)   
and self.base\_Class.general->forAll(g|g.stereotypedBy('AugmentationType'))   
and (  
 self.base\_Class.\_directedRelationshipOfTarget->forAll(r|r.oclIsKindOf(Generalization) implies r.oclAsType(Generalization).specific.stereotypedBy('AugmentationType'))  
)

###### NDR [Rule 7-48]

**[Rule 7-48] (REF, SUB, EXT)** Within the schema, an augmentation element definition:

1. SHALL have a type that is an augmentation type.

2. SHALL use the substitutionGroup attribute such that it is transitively substitutable for the element structures:Augmentation.

An element that is not an augmentation element SHALL NOT meet either of the above criteria.

**Rationale** An augmentation is trivially identifiable as such. The use of the common structures:Augmentation element allows message builders to optionally delay specifying augmentations to be applied to a type until runtime.

[English]

The constraint is enforced by the transformation from PSM to XSD Schema artifact. A property whose type is an «AugmentationType» is an augmentation element. The property may directly or indirectly use the UML subsettedProperty mechanism to identify a substitutionGroup, which will be transitively substitutable for the element structures:Augmentation.

###### NDR3 [Rule 10-30] (INS). Element within instance of augmentation type modifies base

[Rule 10-30](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-30), Element within instance of augmentation type modifies base (INS): [Section 10.4.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.4), Augmentation types

[English]

The instance rule is outside the scope of the NIEM-UML model.

###### NDR3 [Rule 10-31] (REF,EXT). Only an augmentation type name ends in "AugmentationType"

[Rule 10-31](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-31), Only an augmentation type name ends in AugmentationType (REF, EXT): [Section 10.4.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.4), Augmentation types

**[OCL] context** AugmentationType **inv:**

self.name.endsWith('AugmentationType') = self.stereotypedBy('AugmentationType')

###### NDR3 [Rule 10-32] (REF,EXT). Schema component with name ending in "AugmentationType" is an augmentation type

[Rule 10-32](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-32), Schema component with name ending in AugmentationType is an augmentation type (REF, EXT): [Section 10.4.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.4), Augmentation types

**[OCL] context** AugmentationType **inv:**

self.general  
->union(self.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
->forAll(g|g.stereotypedBy('AugmentationType'))

###### NDR3 [Rule 10-33] (REF,EXT). Type derived from augmentation type is an augmentation type

[Rule 10-33](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-33), Type derived from augmentation type is an augmentation type (REF, EXT): [Section 10.4.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.4), Augmentation types

**[OCL] context** AugmentationType **inv:**

self.\_directedRelationshipOfTarget->select(d|d.oclIsKindOf(Generalization)).oclAsType(Generalization).specific  
->union(self.supplierDependency->select(d|d.stereotypedBy('Restriction')).client->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
->forAll(g|g.stereotypedBy('AugmentationType'))

### <Stereotype> Choice

##### Description

A Choice Class groups a set of attributes whose values are mutually exclusive. That is, in any instance of a Choice Class, at most one of its attributes may be non-empty. Choice represents the use of a choice model group in XML Schema. Section 3.8 of [XML Schema Structures](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/) addresses choice model groups in XML Schema. See [NIEM-NDR] Sections [9.3.1.2,](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.1.2) *Choice* and [9.3.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.2.2), *Choice cardinality*.

##### Extends

UML::Class

##### Constraints

###### no generalizations or subtypes

A Choice Class shall not participate in any Generalizations, either as the general or the special Classifier.

**[OCL] context** Choice **inv:**

self.base\_Class.generalization->isEmpty()   
and self.base\_Class.\_directedRelationshipOfTarget->select(d|d.oclIsKindOf(Generalization))->isEmpty()

###### ownedAttributes have multiplicity 0..1

The ownedAttributes of a Choice class shall have multiplicity 0..1.

**[OCL] context** Choice **inv:**

self.base\_Class.attribute->forAll(a|(a.lower=0) and  
 (a.upper=1))

### <Stereotype> Deprecated

##### Description

A deprecated component is one whose use is not recommended. A deprecated component may be kept in a schema for support of older versions but should not be used in new efforts. A deprecated component may be removed, replaced, or renamed in a later version of a namespace. See [NIEM-NDR] [Section 10.9.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1.1), *Deprecation*.

##### Extends

UML::NamedElement

##### Constraints

###### NDR3 [Rule 10-66] (REF,EXT). Component marked as deprecated is deprecated component

[Rule 10-66](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-66), Component marked as deprecated is deprecated component (REF, EXT): [Section 10.9.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1.1), Deprecation

[English]

Rule is informative. Provisioning ensure that an appinfo:deprecated maps to a model element stereotyped by Deprecated.

### <Stereotype> Documentation

##### Description

A Documentation Comment is the data definition of the Element that owns it.  For an Element owning only one Comment, that Comment will be inferred to be a Documentation Comment. A Documentation Comment owned by an Element representing a NIEM type or property is implemented as a documentation element of the annotation for the corresponding type definition or property declaration.

##### Extends

UML::Comment

##### Constraints

###### max one «Documentation» per Element

The owner of a Documentation Comment must have no other Documentation Comments.

**[OCL] context** Documentation **inv:**

self.base\_Comment.annotatedElement->notEmpty() and  
 self.base\_Comment.annotatedElement->forAll(e|e=self.base\_Comment.owningElement) and  
 (self.base\_Comment.owningElement.ownedComment->select(c|c.stereotypedBy('Documentation'))->size()=1)

### <Stereotype> List

##### Description

A List is a DataType whose values consist of a finite length (possibly empty) sequence of values of another DataType, which is the item type of the List. A List DataType must have a single Property with multiplicity 0..\* whose type is the item type. The name of this element is not material.  A List DataType is implemented in XML schema as a list simple type definition. List represents a relationship between two simple type definitions: the first is a list simple type definition whose item type definition is the second. This relationship is implemented in XML Schema through the itemType attribute on the xsd:list element of the list simple type definition, the actual value of which resolves to the second type definition. Section 3.14 of [XML Schema Structures](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/) addresses list simple type definitions in XML Schema. See [NIEM-NDR] Sections [9.1.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.1), *Simple types prohibited as list item types* and [11.1.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1.2.1), *Derivation by list*.

##### Extends

UML::DataType

##### Constraints

###### NDR3 [Rule 11-6] (REF,EXT). Use lists only when data is uniform

[Rule 11-6](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-6), Use lists only when data is uniform (REF, EXT): [Section 11.1.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1.2.1), Derivation by list

[English]

Not currently expressed in OCL.

###### NDR3 [Rule 11-7] (REF,EXT). List item type defined by conformant schemas

[Rule 11-7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-7), List item type defined by conformant schemas (REF, EXT): [Section 11.1.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1.2.1), Derivation by list

[English]

self.base\_DataType.attribute.type  
->forEach(t|t.\_'package'->forAll(p|p.appliedStereotype('Namespace').oclAsType(NIEM\_UML\_Profile::NIEM\_Common\_Profile::Namespace).isConformant or( p.name='XMLPrimitiveTypes'))

###### NDR3 [Rule 9-13](REF,EXT). No list item type of xs:ID

[Rule 9-13](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-13), No list item type of xs:ID (REF, EXT): [Section 9.1.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.1), Simple types prohibited as list item types

**[OCL] context** List **inv:**

self.base\_DataType.attribute.type->exists(t|not((t.name='ID')and(t.\_'package'.name='XMLPrimitiveTypes')))

###### NDR3 [Rule 9-14] (REF,EXT). No list item type of xs:IDREF

[Rule 9-14](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-14), No list item type of xs:IDREF (REF, EXT): [Section 9.1.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.1), Simple types prohibited as list item types

**[OCL] context** List **inv:**

self.base\_DataType.attribute.type->exists(t|not((t.name='IDREF')and(t.\_'package'.name='XMLPrimitiveTypes')))

###### NDR3 [Rule 9-15] (REF,EXT). No list item type of xs:anySimpleType

[Rule 9-15](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-15), No list item type of xs:anySimpleType (REF, EXT): [Section 9.1.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.1), Simple types prohibited as list item types

**[OCL] context** List **inv:**

self.base\_DataType.attribute.type->exists(t|not((t.name='anySimpleType')and(t.\_'package'.name='XMLPrimitiveTypes')))

###### NDR3 [Rule 9-16] (REF,EXT). No list item type of xs:ENTITY

[Rule 9-16](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-16), No list item type of xs:ENTITY (REF, EXT): [Section 9.1.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.1), Simple types prohibited as list item types

**[OCL] context** List **inv:**

self.base\_DataType.attribute.type->exists(t|not((t.name='ENTITY')and(t.\_'package'.name='XMLPrimitiveTypes')))

###### no generalizations

A List shall not have any generalizations.

**[OCL] context** List **inv:**

self.base\_DataType.generalization->isEmpty()

###### single ownedAttribute with multiplicity 0..\* typed DataType

A List DataType shall have a single ownedAttribute with multiplicity 0..\* whose type is also a DataType.

**[OCL] context** List **inv:**

(self.base\_DataType.attribute->size()=1)   
and  
 self.base\_DataType.attribute ->forAll(a|(a.lower=0) and (a.upper=-1))

### <Stereotype> LocalTerm

##### Description

The LocalTerm stereotype defines a domain-specific word, phrase, acronym, or other string of characters used in a LocalVocabulary. It may occur as a term within the name of a schema component within the schema document.  The domain-specific term is represented by the EnumerationLiteral’s name. NDR SourceText is represented as UML ownedComment.body.  See [NIEM-NDR] [Section 10.8.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.2.1), *Use of Acronyms, Initialisms, Abbreviations, and Jargon*

##### Extends

UML::EnumerationLiteral

##### Properties

###### definition : String [0..1]

 The value of definition is a dictionary-style description of the meaning of the local term.

###### literal : String [0..1]

 The value of literal is the meaning of the local term, provided as a full, plain-text form of the term. This may be useful when a local term is an abbreviation, acronym, or diminutive form of a longer term.

###### sourceURIs : String [0..\*]

The value of sourceURIs is a list of URIs, each of which is an identifier or locator for an originating or authoritative document defining the term.

##### Constraints

###### NDR3 [Rule 10-74] (REF,EXT). term:LocalTerm annotates schema

[Rule 10-74](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-74), term:LocalTerm annotates schema (REF, EXT): [Section 10.9.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.2), The NIEM local terminology namespace

[English]

The constraint is realized through provisioning:  
A provisioned term:LocalTerm will be owned by an xs:appinfo which is owned by an xs:annotation which is owned by an xs:schema

###### NDR3 [Rule 10-75] (REF,EXT). term:LocalTerm has literal or definition

[Rule 10-75](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-75), term:LocalTerm has literal or definition (REF, EXT): [Section 10.9.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.2), The NIEM local terminology namespace

**[OCL] context** LocalTerm **inv:**

self.literal.oclIsUndefined()<>self.definition.oclIsUndefined()

### <Stereotype> LocalVocabulary

##### Description

Local vocabulary defines a set of domain specific terms or abbreviations that then may be used in NIEM names and definitions. The local vocabulary is defined as a stereotype of Enumeration where each EnumerationLliteral is a vocabulary term represented by the «LocalTerm» stereotype. See [NIEM-NDR] [Section 10.8.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.2.1), *Use of Acronyms, Initialisms, Abbreviations, and Jargon*.

##### Extends

UML::Enumeration

### <Stereotype> MetadataApplication

##### Description

The «MetadataApplication» stereotype applies to a Usage between a «MetadataType» Class and either another «MetadataType» Class or a Property. It represents a constraint on a NIEM «MetadataType» that limits the application of the NIEM «MetadataType» to specific schema types or schema elements. If a «MetadataType» Class is the client of a «MetadataApplication» Usage, then any Property with the «MetadataType» Class as its type must be for a Class that is a (direct or indirect) subclass of the supplier Class of the «MetadataApplication». A «MetadataType» Class may be the client of multiple «MetadataApplication» Usages, in which case a Property for it may be in a Class that is a subclass of a supplier Class of any of the «MetadataApplication»s. If a «MetadataType» is not a client of any «MetadataApplication», then it applies to any type. If a Property is the supplier of a «MetadataApplication» Usage, then the allowable elements referencing the «MetadataType» are restricted to the indicator supplier Property and any of its (transitive) substitutions.  A «MetadataApplication» Usage with a Class supplier is implemented in NIEM as appinfo:appliesToTypes.  A «MetadataApplication» Usage with a Property supplier is implemented in NIEM as appinfo:appliesToElements.  See [NIEM-NDR] [Section 10.9.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1.2), *appinfo:appliesToTypes annotation*, and [Section 10.9.1.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1.3), *appinfo:appliesToElements annotation*.

##### Extends

UML::Usage

##### Constraints

###### NDR3 [Rule 10-70] (REF,EXT). appinfo:appliesToTypes annotates metadata element

[Rule 10-70](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-70), appinfo:appliesToTypes annotates metadata element (REF, EXT): [Section 10.9.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1.2), appinfo:appliesToTypes annotation

**[OCL] context** MetadataApplication **inv:**

self.base\_Usage.client->forAll(g|g.oclIsKindOf(Property) and g.oclAsType(Property).type.stereotypedBy('MetadataType'))

###### NDR3 [Rule 10-71] (SET). appinfo:appliesToTypes references types

[Rule 10-71](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-71), appinfo:appliesToTypes references types (SET): [Section 10.9.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1.2), appinfo:appliesToTypes annotation

**[OCL] context** MetadataApplication **inv:**

self.base\_Usage.supplier->forAll(g|g.oclIsKindOf(Classifier)or g.oclIsKindOf(Property))

###### NDR3 [Rule 10-72] (REF,EXT). appinfo:appliesToElements annotates metadata element

[Rule 10-72](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-72), appinfo:appliesToElements annotates metadata element (REF, EXT): [Section 10.9.1.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1.3), appinfo:appliesToElements annotation

**[OCL] context** MetadataApplication **inv:**

self.base\_Usage.client->forAll(g|g.oclIsKindOf(Property) and g.oclAsType(Property).type.stereotypedBy('MetadataType'))

###### NDR3 [Rule 10-73] (SET). appinfo:appliesToElements references elements

[Rule 10-73](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-73), appinfo:appliesToElements references elements (SET): [Section 10.9.1.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1.3), appinfo:appliesToElements annotation

**[OCL] context** MetadataApplication **inv:**

self.base\_Usage.supplier->forAll(g|g.oclIsKindOf(Classifier)or g.oclIsKindOf(Property))

### <Stereotype> MetadataType

##### Description

A MetadataType is a NIEMType Class that represents a NIEM metadata type. A NIEM metadata type describes data about data, that is, information that is not descriptive of objects and their relationships, but is descriptive of the data itself. Metadata is specified as an instance of a metadata type and may include information such as the security of a piece of data or the source of the data. The applicability of such metadata may be modeled using MetadataApplication dependencies to one or more classes representing the applicable types.

MetadataType is implemented in XML Schema as a complex type definition with complex content. See [NIEM-NDR] [Section 10.5.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.5.1), *Metadata types*.

##### Generalization

[NIEMType](#_cddcf0aa38f9fb92183a65a83b2b548f)

##### Constraints

###### NDR3 [Rule 10-36] (REF,EXT). Metadata type has data about data

[Rule 10-36](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-36), Metadata type has data about data (REF, EXT): [Section 10.5.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.5.1), Metadata types

[English]

Rule is definitional.

###### NDR3 [Rule 10-37] (REF,EXT). Metadata type derived from structures:MetadataType

[Rule 10-37](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-37), Metadata type derived from structures:MetadataType (REF, EXT): [Section 10.5.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.5.1), Metadata types

**[OCL] context** MetadataType **inv:**

self.base\_Class.general->isEmpty()  
and  
self.base\_Class.clientDependency->select(d|d.stereotypedBy('Restriction'))->isEmpty()

###### NDR3 [Rule 10-38] (REF,EXT). Metadata types are derived from metadata types

[Rule 10-38](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-38), Metadata types are derived from metadata types (REF, EXT): [Section 10.5.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.5.1), Metadata types

**[OCL] context** MetadataType **inv:**

self.base\_Class.general->isEmpty()  
and  
self.base\_Class.clientDependency->select(d|d.stereotypedBy('Restriction'))->isEmpty()

### <Stereotype> Namespace

##### Description

A Namespace Package represents a NIEM namespace identified by a target namespace URI. All UML model elements contained, directly or indirectly within the Package, that represents NIEM types and properties, are considered to be in this target namespace. A Namespace Package is implemented in XML Schema as an XML schema document.

##### Extends

UML::Package

##### Properties

###### conformanceTargets : String [0..\*]

The Conformance Targets Attribute Specification defines an attribute that, when it appears in an XML document, claims the document conforms to one or more conformance targets. This pattern and specification was developed to overcome shortcomings in the NIEM 2 ConformantIndicator element, and to provide needed capabilities in future specifications. The attribute is a claim of conformance, and not a statement that should be trusted by a validating system. A validator would use this claim to identify to which conformance rules a document should be validated. The attribute's value is a list of internationalized resource identifiers (IRIs). A later specification may define an IRI for its conformance target, and when an XML document has that IRI in its conformance target attribute, the document is claiming to conform to that conformance target. The *effective conformance targets attribute* of a conformant document is the first occurrence of the attribute {http://release.niem.gov/niem/conformanceTargets/3.0/}conformanceTargets, in document order.

The only conformance target explicitly defined in NIEM 3 are

<http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/#ReferenceSchemaDocument>

and

<http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/#ExtensionSchemaDocument>

For NIEM-3 UML, the above formally defined conformance targets are implicitly associated with a NIEM target schema based on the defaultPurpose of an «InformationModel». Thus, the "conformanceTargets" tag need be populated only with domain-specific conformance target values.

###### defaultPrefix : String [0..1]

The default prefix for the namespace, used to represent common NIEM prefixes. This prefix should be used on all XML and/or XML Schema serializations using that namespace, unless it conflicts with another XML and/or XML Schema serialization. If there is a conflict, the actual prefix used is the given default prefix with a number appended in order to make it unique.

###### isConformant : Boolean [1]

Indicates whether the namespace is NIEM-conformant.  The targets it conforms to are specified by the defaultPurpose of the related «InformationModel», and by its conformanceTargets attribute.

###### targetNamespace : String [1]

The target namespace URI for this NIEM namespace.  It is implemented in XML Schema as the value of the targetNamespace attribute on the xsd:schema document element. Per Rules [9-82](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-82) and [9-83](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-83) of [NIEM-NDR], the value of the targetNamespace attribute must be present and must be an absolute URI.

###### version : String [1]

The version of the NIEM namespace. It is implemented in XML Schema as the value of the version attribute on the xsd:schema document element. Per [Rule 9-84](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-84) of [NIEM-NDR], the value of the version attribute must be present and must not be the empty string. Default is "1".

##### Constraints

###### NDR3 [Rule 10-7] (REF,EXT). Import of external namespace has data definition

[Rule 10-7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-7), Import of external namespace has data definition (REF, EXT): [Section 10.2.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.1), Import of external namespace.

[English]

This constraint resolved during provisioning;   
in UML, all InformationModels must be documented;   
an xs:import is generated if content of the InformationModel refers to another InformationModel;  
generated xs:import elements from a external namespace will include xs:documentation obtained from their documented "external" InformationModels.

###### NDR3 [Rule 7-2] (REF,EXT,INS). Document uses XML namespaces properly

[Rule 7-2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_7-2), Document uses XML namespaces properly (REF, EXT, INS): [Section 7.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_7.2), Conformance to XML Namespaces

[English]

self.targetNamespace is namespace-well-formed and namespace-valid.  
  
namespace-valid conformance is enforced during provisioning by ensuring that the production of names is conformant with NDR naming rules and XML Namespaces Specification.  
namespace-well-formed conformance is enforced during provisioning by ensuring that the target xml document is well formed with respected to the XML Namespaces Specification.

###### NDR3 [Rule 7-3] (REF,EXT). Document is a schema document

[Rule 7-3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_7-3), Document is a schema document (REF, EXT): [Section 7.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_7.3), Conformance to XML Schema

[English]

Enforced during provisioning to Schema from Namespace Package.

###### NDR3 [Rule 7-4] (REF,EXT). Document element is xs:schema

[Rule 7-4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_7-4), Document element is xs:schema (REF, EXT): [Section 7.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_7.4), Conformance to XML Schema

[English]

Enforced during provisioning to Schema from Namespace Package.

###### NDR3 [Rule 7-5] (REF,EXT). Component name follows ISO 11179 Part 5 Annex A

[Rule 7-5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_7-5), Component name follows ISO 11179 Part 5 Annex A (REF, EXT): [Section 7.5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_7.5), ISO 11179 Part 5

[English]

The default normative naming rules based on ISO 11179-5 are not easily computable, so are not represented as an executable OCL Constraint.

###### NDR3 [Rule 9-10] (REF,EXT). Simple type definition is top-level

[Rule 9-10](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-10), Simple type definition is top-level (REF, EXT): [Section 9.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2), Simple type definition

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Class)).oclAsType(Class).nestedClassifier->isEmpty()

###### NDR3 [Rule 9-1] (REF,EXT). No base type in the XML namespace

[Rule 9-1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-1), No base type in the XML namespace (REF, EXT): [Section 9.1.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.1.1), Types prohibited as base types

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
 ->forAll(g|g.\_'package'.name<>'xml')  
)

###### NDR3 [Rule 9-23] (REF,EXT). Enumeration has data definition

[Rule 9-23](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-23), Enumeration has data definition (REF, EXT): [Section 9.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2), Simple type definition

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Enumeration)).oclAsType(Enumeration).ownedComment  
->exists(c|not(c.\_'body'.oclIsUndefined()) and (c.\_'body'<>''))

###### NDR3 [Rule 9-24] (REF,EXT). Complex type definitions is top-level

[Rule 9-24](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-24), Complex type definitions is top-level (REF, EXT): [Section 9.1.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3), Complex type definition

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Class)).oclAsType(Class).nestedClassifier->isEmpty()

###### NDR3 [Rule 9-25] (REF,EXT). Complex type has data definition

[Rule 9-25](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-25), Complex type has data definition (REF, EXT): [Section 9.1.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3), Complex type definition

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).ownedComment  
->exists(c|not(c.\_'body'.oclIsUndefined()) and (c.\_'body'<>''))

###### NDR3 [Rule 9-26] (REF,EXT). No mixed content on complex type

[Rule 9-26](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-26), No mixed content on complex type (REF, EXT): [Section 9.1.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3.1), No mixed content

[English]

There is no option in NIEM-UML to specify mixed content, consequently there is no mixed content produced during provisioning of target schemas.

###### NDR3 [Rule 9-27] (REF,EXT). No mixed content on complex content

[Rule 9-27](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-27), No mixed content on complex content (REF, EXT): [Section 9.1.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3.1), No mixed content

[English]

There is no option in NIEM-UML to specify mixed content, consequently there is no mixed content produced during provisioning of target schemas.

###### NDR3 [Rule 9-28] (REF,EXT). Complex type content is explicitly simple or complex

[Rule 9-28](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-28), Complex type content is explicitly simple or complex (REF, EXT): [Section 9.1.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3), Complex type definition

[English]

Complex type content is always enforced to be simple or complex based on defined provisioning.

###### NDR3 [Rule 9-2] (REF,EXT). No base type of xs:ID

[Rule 9-2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-2), No base type of xs:ID (REF, EXT): [Section 9.1.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.1.1), Types prohibited as base types

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
 ->select(t|(t.name='ID') and (t.\_'package'.name='XMLPrimitiveTypes'))->size()=0  
)

###### NDR3 [Rule 9-3] (REF,EXT). No base type of xs:IDREF

[Rule 9-3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-3), No base type of xs:IDREF (REF, EXT): [Section 9.1.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.1.1), Types prohibited as base types

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
 ->select(t|(t.name='IDREF') and (t.\_'package'.name='XMLPrimitiveTypes'))->size()=0  
)

###### NDR3 [Rule 9-40] (REF,EXT). Element type not in the XML namespace

[Rule 9-40](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-40), Element type not in the XML namespace (REF, EXT): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).attribute.type->forAll(c|  
 c.\_'package'.name<>'xml'  
)

###### NDR3 [Rule 9-44] (REF,EXT). No element default value

[Rule 9-44](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-44), No element default value (REF, EXT): [Section 9.2.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1.1), No element value constraints

[English]

Constraint is realized via Provisioning, which does not create any @default attributes.

###### NDR3 [Rule 9-45] (REF,EXT). No element fixed value

[Rule 9-45](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-45), No element fixed value (REF, EXT): [Section 9.2.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1.1), No element value constraints

[English]

Constraint is realized via Provisioning, which does not create any @fixed attributes.

###### NDR3 [Rule 9-46] (REF). Element declaration is nillable

[Rule 9-46](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-46), Element declaration is nillable (REF): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

[English]

Specification as OCL Constraint is deferred.

###### NDR3 [Rule 9-4] (REF,EXT). No base type of xs:IDREFS

[Rule 9-4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-4), No base type of xs:IDREFS (REF, EXT): [Section 9.1.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.1.1), Types prohibited as base types

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
 ->select(t|(t.name='IDREFS') and (t.\_'package'.name='XMLPrimitiveTypes'))->size()=0  
)

###### NDR3 [Rule 9-58] (REF,EXT). No use of element xs:notation

[Rule 9-58](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-58), No use of element xs:notation (REF, EXT): [Section 9.2.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.4), Notation declaration

[English]

This constraint enforced by provisioning, which does not produce any xs:notation schema components.

###### NDR3 [Rule 9-59] (EXT). Model group does not affect meaning

[Rule 9-59](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-59), Model group does not affect meaning (EXT): [Section 9.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.1), Model group

[English]

This constraint is not computable.

###### NDR3 [Rule 9-5] (REF,EXT). No base type of xs:anyType

[Rule 9-5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-5), No base type of xs:anyType (REF, EXT): [Section 9.1.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.1.1), Types prohibited as base types

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
 ->select(t|(t.name='anyType') and (t.\_'package'.name='XMLPrimitiveTypes'))->size()=0  
)

###### NDR3 [Rule 9-60] (REF,EXT). No xs:all

[Rule 9-60](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-60), No xs:all (REF, EXT): [Section 9.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.1), Model group

[English]

This constraint enforced by provisioning, there is not model representation for xs:all and no production of an xs:all model group.

###### NDR3 [Rule 9-6] (REF,EXT). No base type of xs:anySimpleType

[Rule 9-6](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-6), No base type of xs:anySimpleType (REF, EXT): [Section 9.1.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.1.1), Types prohibited as base types

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
 ->select(t|(t.name='anySimpleType') and (t.\_'package'.name='XMLPrimitiveTypes'))->size()=0  
)

###### NDR3 [Rule 9-7] (REF,EXT). No base type of xs:NOTATION

[Rule 9-7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-7), No base type of xs:NOTATION (REF, EXT): [Section 9.1.1.1,](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.1.1) Types prohibited as base types

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
 ->select(t|(t.name='NOTATION') and (t.\_'package'.name='XMLPrimitiveTypes'))->size()=0  
)

###### NDR3 [Rule 9-82] (REF,EXT). Schema document defines target namespace

[Rule 9-82](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-82), Schema document defines target namespace (REF, EXT): [Section 9.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.7), Schema as a whole

**[OCL] context** Namespace **inv:**

not(self.targetNamespace.oclIsUndefined()) and  
 (self.targetNamespace<>'')

###### NDR3 [Rule 9-83] (REF,EXT). Target namespace is absolute URI

[Rule 9-83](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-83), Target namespace is absolute URI (REF, EXT): [Section 9.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.7), Schema as a whole

[English]

Specification of this constraint in OCL has been deferred.

###### NDR3 [Rule 9-84] (REF,EXT). Schema has version

[Rule 9-84](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-84), Schema has version (REF, EXT): [Section 9.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.7), Schema as a whole

**[OCL] context** Namespace **inv:**

not(self.version.oclIsUndefined())   
and  
self.version<>''

###### NDR3 [Rule 9-89] (REF,EXT). xs:import must have namespace

[Rule 9-89](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-89), xs:import must have namespace (REF, EXT): [Section 9.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8), Schema assembly

[English]

This constraint resolved during provisioning; all xs:import declarations are implicitly specified by the relationships between Information Model elements and will include the namespace specified in the targetNamespace of the referenced Information Model.

###### NDR3 [Rule 9-8] (REF,EXT). No base type of xs:ENTITY

[Rule 9-8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-8), No base type of xs:ENTITY (REF, EXT): [Section 9.1.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.1.1), Types prohibited as base types

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
 ->select(t|(t.name='ENTITY') and (t.\_'package'.name='XMLPrimitiveTypes'))->size()=0  
)

###### NDR3 [Rule 9-90] (SET). XML Schema document set must be complete

[Rule 9-90](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-90), XML Schema document set must be complete (SET): [Section 9.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8), Schema assembly

[English]

This constraint resolved during provisioning; the transitive closure of all schemas referenced will be in the schema document set for an MPD.

###### NDR3 [Rule 9-91] (REF,EXT). Namespace referenced by attribute type is imported

[Rule 9-91](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-91), Namespace referenced by attribute type is imported (REF, EXT): [Section 9.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8.1), Namespaces for referenced components are imported

[English]

This constraint resolved during provisioning; the transitive closure of all schemas referenced will be in the schema document set for an MPD.

###### NDR3 [Rule 9-92] (REF,EXT). Namespace referenced by attribute base is imported

[Rule 9-92](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-92), Namespace referenced by attribute base is imported (REF, EXT): [Section 9.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8.1), Namespaces for referenced components are imported

[English]

This constraint resolved during provisioning; the transitive closure of all schemas referenced will be in the schema document set for an MPD.

###### NDR3 [Rule 9-93] (REF,EXT). Namespace referenced by attribute itemType is imported

[Rule 9-93](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-93), Namespace referenced by attribute itemType is imported (REF, EXT): [Section 9.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8.1), Namespaces for referenced components are imported

[English]

This constraint resolved during provisioning; the transitive closure of all schemas referenced will be in the schema document set for an MPD.

###### NDR3 [Rule 9-94] (REF,EXT). Namespaces referenced by attribute memberTypes is imported

[Rule 9-94](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-94), Namespaces referenced by attribute memberTypes is imported (REF, EXT): [Section 9.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8.1), Namespaces for referenced components are imported

[English]

This constraint resolved during provisioning; the transitive closure of all schemas referenced will be in the schema document set for an MPD.

###### NDR3 [Rule 9-95] (REF,EXT). Namespace referenced by attribute ref is imported

[Rule 9-95](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-95), Namespace referenced by attribute ref is imported (REF, EXT): [Section 9.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8.1), Namespaces for referenced components are imported

[English]

This constraint resolved during provisioning; the transitive closure of all schemas referenced will be in the schema document set for an MPD.

###### NDR3 [Rule 9-96] (REF,EXT). Namespace referenced by attribute substitutionGroup is imported

[Rule 9-96](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-96), Namespace referenced by attribute substitutionGroup is imported (REF, EXT): [Section 9.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8.1), Namespaces for referenced components are imported

[English]

This constraint resolved during provisioning; the transitive closure of all schemas referenced will be in the schema document set for an MPD.

###### NDR3 [Rule 9-9] (REF,EXT). No base type of xs:ENTITIES

[Rule 9-9](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-9), No base type of xs:ENTITIES (REF, EXT): [Section 9.1.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.1.1), Types prohibited as base types

**[OCL] context** Namespace **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
 ->select(t|(t.name='ENTITIES') and (t.\_'package'.name='XMLPrimitiveTypes'))->size()=0  
)

### <Stereotype> NIEMType

##### Description

A NIEMType is a Class that represents one of the specific semantic kinds of NIEM complex types (i.e., types that may have attributive structure). NIEMType is abstract.

##### Extends

UML::Class

### <Stereotype> ObjectType

##### Description

ObjectType is a NIEMType Class that represents a NIEM object type. A NIEM object type represents some kind of object: a thing with its own lifespan that has some existence. The object may or may not be a physical object. It may be a conceptual object.  ObjectType is implemented in XML Schema as a complex type definition. Section 3.4 of [XML Schema Structures](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/) addresses complex type definitions in XML Schema. See [NIEM-NDR] [Section 10.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.1), *General object types*.

##### Generalization

[NIEMType](#_cddcf0aa38f9fb92183a65a83b2b548f)

##### Constraints

###### NDR [Rule 7-39] (REF, EXT)

Within the schema, an object type SHALL be a complex type definition that either constitutes a NIEM-conformant component or for which there exists a NIEM-conformant component of one of the following forms:

* Has simple content, is based on a simple type, and contains the attribute group structures:SimpleObjectAttributeGroup, and has application information appinfo:Base of structures:Object.
* Has complex content, and is based on complex type structures:ComplexObjectType, and has application information appinfo:Base of structures:Object.
* Is a complex type that is derived from an object type, which is defined according to this rule.

###### NDR3 [Rule 10-18] (REF,EXT). Proxy type has designated structure

[Rule 10-18](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-18), Proxy type has designated structure (REF, EXT): [Section 10.2.5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.5), Proxy types

**[OCL] context** ObjectType **inv:**

(  
 self.general->forAll(g|(g.\_'package'.name='XMLPrimitiveTypes')and(g.name=self.name))  
 and  
 self.clientDependency->select(d|d.stereotypedBy('XSDSimpleContent')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(g|(g.\_'package'.name='XMLPrimitiveTypes')and(g.name=self.name))  
)  
implies  
self.attribute->isEmpty()

###### NDR3 [Rule 10-21] (REF). Augmentable type has augmentation point element

[Rule 10-21](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-21), Augmentable type has augmentation point element (REF): [Section 10.4.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.1), Augmentable types

[English]

The constraint is enforced by provisioning; if an AugmentationPoint is not defined in the model, then it is created

###### NDR3 [Rule 10-22] (REF,EXT). Augmentable type has at most one augmentation point element

[Rule 10-22](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-22), Augmentable type has at most one augmentation point element (REF, EXT): [Section 10.4.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.1), Augmentable types

**[OCL] context** ObjectType **inv:**

self.attribute->select(a|self.name.replace('Type','AugmentationPoint')=a.name)->size()<=1

###### NDR3 [Rule 10-23] (REF,EXT). Augmentation point corresponds to augmentable type

[Rule 10-23](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-23), Augmentation point corresponds to augmentable type (REF, EXT): [Section 10.4.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.2), Augmentation point element declarations

**[OCL] context** ObjectType **inv:**

self.attribute->select(a|a.name.endsWith('AugmentationPoint'))  
->forAll(a|self.\_'package'.ownedType.name->exists(n|n.replace('Type','AugmentationPoint')=a.name))

###### NDR3 [Rule 10-24] (REF,EXT). An augmentation point has no type

[Rule 10-24](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-24), An augmentation point has no type (REF, EXT): [Section 10.4.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.2), Augmentation point element declarations

**[OCL] context** ObjectType **inv:**

self.attribute->select(a|a.name.endsWith('AugmentationPoint'))  
->forAll(a|a.type.oclIsUndefined())

###### NDR3 [Rule 10-25] (REF,EXT). An augmentation point has no substitution group

[Rule 10-25](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-25), An augmentation point has no substitution group (REF, EXT): [Section 10.4.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.2), Augmentation point element declarations

**[OCL] context** ObjectType **inv:**

self.attribute->select(a|a.name.endsWith('AugmentationPoint'))  
->forAll(a|a.subsettedProperty->isEmpty())

###### NDR3 [Rule 10-26] (REF,EXT). Augmentation point element may only be referenced by its type

[Rule 10-26](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-26), Augmentation point element may only be referenced by its type (REF, EXT): [Section 10.4.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.3), Augmentation point element use

**[OCL] context** ObjectType **inv:**

not(self.stereotypedBy('PropertyHolder'))  
implies  
self.attribute->select(a|a.name.endsWith('AugmentationPoint'))  
->forAll(a|a.name.replace('AugmentationPoint','Type')=self.name)

###### NDR3 [Rule 10-27] (REF). Augmentation point reference is optional

[Rule 10-27](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-27), Augmentation point reference is optional (REF): [Section 10.4.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.3), Augmentation point element use

**[OCL] context** ObjectType **inv:**

(  
 not(self.\_'package'.oclIsUndefined())  
 and  
 self.\_'package'.stereotypedBy('InformationModel')  
 and  
 self.\_'package'.appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel)  
 ->forAll(im|(im.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode.subset)or(im.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode.reference))  
)   
implies  
self.attribute->select(a|a.name.endsWith('AugmentationPoint'))  
->forAll(a|a.lower=0)

###### NDR3 [Rule 10-28] (REF). Augmentation point reference is unbounded

[Rule 10-28](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-28), Augmentation point reference is unbounded (REF): [Section 10.4.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.3), Augmentation point element use

**[OCL] context** ObjectType **inv:**

(  
 not(self.\_'package'.oclIsUndefined())  
 and  
 self.\_'package'.stereotypedBy('InformationModel')  
 and  
 self.\_'package'.appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel)  
 ->forAll(im|  
 (im.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode.subset) or  
 (im.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode.reference))  
)   
implies  
self.attribute->select(a|a.name.endsWith('AugmentationPoint'))  
->forAll(a|a.upper<0)

###### NDR3 [Rule 10-29] (REF). Augmentation point reference must be last particle

[Rule 10-29](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-29), Augmentation point reference must be last particle (REF, EXT): [Section 10.4.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.3), Augmentation point element use

**[OCL] context** ObjectType **inv:**

(  
 not(self.\_'package'.oclIsUndefined())  
 and  
 self.\_'package'.stereptypedBy('informationModel')  
 and  
 self.\_'package'.appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel)  
 ->forAll(im|  
 (im.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode.subset) or   
 (im.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode.reference))  
)   
implies  
self.attribute->select(a|a.name.endsWith('AugmentationPoint'))  
->forAll(a|self.attribute->last()=a)

###### NDR3 [Rule 11-11] (REF,EXT). Complex type with simple content has structures:SimpleObjectAttributeGroup

[Rule 11-11](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-11), Complex type with simple content has structures:SimpleObjectAttributeGroup (REF, EXT): [Section 11.1.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1.3), Complex type definition

[English]

This constraint realized during provisioning;  
The structures:SimpleObjectAttributeGroup is not in the UML-NIEM model, it is produced as required during construction of a complex type with simple content

###### NDR3 [Rule 11-1] (REF,EXT). Name of type ends in "Type"

[Rule 11-1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-1), Name of type ends in Type (REF, EXT): [Section 11.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1), Type definition components

**[OCL] context** ObjectType **inv:**

(  
 self.\_'package'.stereotypedBy('InformationModel')  
 and  
 not(self.stereotypedBy('PropertyHolder'))  
 and  
 not(self.stereotypedBy('LocalVocabulary')   
 and  
 not(self.general->forAll(g|(g.name=self.name)and(g.\_'package'.name='XMLPrimitiveTypes'))  
)  
implies  
self.name.endsWith('Type')

###### NDR3 [Rule 11-2] (REF,EXT). Name of type other than proxy type is in upper camel case

[Rule 11-2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-2), Name of type other than proxy type is in upper camel case (REF, EXT): [Section 11.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1), Type definition components

**[OCL] context** ObjectType **inv:**

(  
 self.\_'package'.stereotypedBy('InformationModel')  
 and  
 not(self.stereotypedBy('PropertyHolder'))  
 and  
 not(self.stereotypedBy('LocalVocabulary')   
 and  
 not(self.general->forAll(g|(g.name=self.name)and(g.\_'package'.name='XMLPrimitiveTypes'))  
)  
implies  
self.name.match('^([A-Z][A-Za-z0-9\\-]\*)+$')

###### NDR3 [Rule 11-32] (REF,EXT). Standard opening phrase for complex type

[Rule 11-32](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-32), Standard opening phrase for complex type (REF, EXT): [Section 11.6.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.6.1.1), Data definition opening phrases

**[OCL] context** ObjectType **inv:**

(  
 not(self.stereotypedBy('LocalVocabulary'))   
 and  
 not(self.stereotypedBy('PropertyHolder'))   
 and  
 self.namespace.stereotypedBy('InformationModel')   
 )   
implies   
if (self.name.endsWith('AssociationType')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^a data type for (a relationship|an association)'))   
 else if (self.name.endsWith('AugmentationType')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^a data type (that supplements|for additional information about)'))   
 else if (self.name.endsWith('MetadataType')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^a data type for (metadata about|information that further qualifies)'))   
 else self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^a data type'))   
 endif   
 endif   
endif

###### NDR3 [Rule 11-3] REF,EXT. Base type definition defined by conformant schema

[Rule 11-3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-3), Base type definition defined by conformant schema (REF, EXT): [Section 11.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1.1), Type definition hierarchy

**[OCL] context** ObjectType **inv:**

self.general->select(g|not(g.\_'package'.name='XMLPrimitiveTypes'))  
->forAll(g|g.appliedStereotype(NIEM\_UML\_Profile::NIEM\_Common\_Profile::Namespace).isConformant)

###### NDR3 [Rule 11-4] (REF,EXT). Name of simple type ends in "SimpleType"

[Rule 11-4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-4), Name of simple type ends in SimpleType (REF, EXT): [Section 11.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1.2), Simple type definition

**[OCL] context** ObjectType **inv:**

(  
 (self.oclIsKindOf(Enumeration)and self.oclAsType (Enumeration).ownedLiteral->notEmpty())  
 or  
 self.stereotypedBy('List')  
 or  
 self.stereotypedBy('Union')  
 or  
 self.stereotypedBy('ValueRestriction')  
 or  
 self.stereotypedBy('XSDRepresentationRestriction')   
 or  
 self.supplierDependency->exists(c|c.stereotypedBy('XSDSimpleContent'))  
)  
implies  
self.name.endsWith('SimpleType')

###### NDR3 [Rule 11-5] (REF,EXT). Name of simple type is upper camel case

[Rule 11-5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-5), Name of simple type is upper camel case (REF, EXT): [Section 11.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1.2), Simple type definition

**[OCL] context** ObjectType **inv:**

(  
 (self.oclIsKindOf(Enumeration)and self.oclAsType (Enumeration).ownedLiteral->notEmpty())  
 or  
 self.stereotypedBy('List')  
 or  
 self.stereotypedBy('Union')  
 or  
 self.stereotypedBy('ValueRestriction')  
 or  
 self.stereotypedBy('XSDRepresentationRestriction')   
 or  
 self.supplierDependency->exists(c|c.stereotypedBy('XSDSimpleContent'))  
)  
implies  
self.name.match('^([A-Z][A-Za-z0-9\\-]\*)+$')

### <Stereotype> PropertyHolder

##### Description

PropertyHolder is a Class holding global Properties that are not the subject of any specific NIEM type. A Property of a NIEM type may then be defined by reference to a Property of a PropertyHolder by using a References realization with the Property in the PropertyHolder as the supplier. Note that the multiplicity of Properties in a PropertyHolder is immaterial -- the multiplicities are established by Properties in the corresponding References client. The target namespace of Properties in a PropertyHolder is the target namespace of the Namespace Package that contains the PropertyHolder (which may be different than the target namespace of NIEM types that use the Properties in the PropertyHolder). PropertyHolder does not represent any NIEM concept; it exists to permit the user to define a NIEM property that is not the subject of any NIEM type. There are significant differences between the UML representation and XML Schema implementation of a NIEM property. Sections [9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1) and [9.2.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3) of [NIEM-NDR], Rule [9-35](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-35) and Rule [9-47](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-47/), require that an attribute or element declaration be a top-level declaration, but [NIEM-NDR] does not require a corresponding attribute use or element particle; however, Section 7.3.44 of [UML] requires that a Property be the ownedAttribute of a Classifier. Thus in the UML representation, the declaration and use of a Property are not distinct, and the declaration of a Property requires its use. In the XML Schema implementation, the declaration and use are distinct, and the declaration does not require a corresponding use. To resolve this difference, any Property within a PropertyHolder shall represent an attribute or element declaration without a corresponding attribute use or element particle. PropertyHolders may be used to hold the properties of a substitution group. Where a PropertyHolder is used to define a substitution group an extension of that substitution group shall be a subclass of the substitution group PropertyHolder.

##### Extends

UML::Class

### <Stereotype> References

##### Description

The References Stereotype applies to a Realization between Properties, Classes or Packages. It allows for Properties in one Class to be defined by reference to Properties in another class. A References Realization between two classes is defined to be equivalent to having References Realizations between matching Properties of the Classes where matching is determined by identical NIEM names. A References Realization between two packages is defined to be equivalent to having References Realizations between matching Classes contained in the Packages where matching is determined by having identical NIEM names. Matching is based on the NIEMName of the elements, either as derived implicitly or as set explicitly using the ReferenceName stereotype. If a Property is the client of a References Realization, then it represents a NIEM property defined by reference to the NIEM property declaration represented by the supplier of the Realization. It is implemented in XSD schema as an attribute use or element particle that references the attribute or element declaration that implements the supplier of the Realization. Note that the supplier Property may be in a different Namespace than the client property, in which case the attribute or element declaration represented by the supplier will be in a different target namespace than the use represented by the client.

##### Extends

UML::Realization

##### Constraints

###### MPD [Rule 3-02]

**[Rule 3-2]** NIEM subsets may omit elements with zero cardinality and adjust the cardinality of elements in reference schemas from which they are derived, as long as the subset property is maintained.

**[OCL] context** References **inv:**

(  
 ( self.base\_Realization.client->size()=1)   
 and ( self.base\_Realization.supplier->size()=1)   
 and self.base\_Realization.client->forAll(client|client.oclIsKindOf(Classifier))  
 and self.base\_Realization.supplier->forAll(supplier|supplier.oclIsKindOf(Classifier)and not(supplier.stereotypedBy('PropertyHolder')) )  
 ) implies (   
 (  
 self.base\_Realization.client.oclAsType(Classifier).attribute  
 ->forAll(clientAttribute| self.base\_Realization.supplier.oclAsType(Classifier).attribute  
 ->forAll(supplierAttribute|  
 (clientAttribute.name=supplierAttribute.name)  
 implies   
 (   
 (clientAttribute.lower>=supplierAttribute.lower)   
 and   
 ( (supplierAttribute.upper=-1) or (clientAttribute.upper<=supplierAttribute.upper) )   
 and ( (clientAttribute.upper=-1) or (clientAttribute.lower<=clientAttribute.upper) )  
 )  
 )  
 )  
 )   
 and  
 ( self.base\_Realization.supplier.oclAsType(Classifier).attribute->select(a|a.lower>0)  
 ->forAll(supplierAttribute| self.base\_Realization.client.oclAsType(Classifier).attribute->exists(clientAttribute|clientAttribute.name=supplierAttribute.name) )  
 )   
)

###### References must be between like metaclasses

References may only be between packages, classifiers or properties and the metatype of the client must be the same as the metatype of the supplier.

### <Stereotype> Representation

##### Description

One need frequently faced by schema developers is for multiple representations for a single concept. To handle this need, NIEM has adopted the Representation pattern, in which a type may contain a representation element, and the various representations for that element type are in the substitution group for that representation element. In NIEM-3 UML, the Representation concept may be expressed as an abstract type-less Property whose name has a suffix of "Representation".  Alternatively, an abstract type-less Property Stereotyped by «Representation» may be used to represent the Representation concept, in which case the NIEM naming rule for Representation elements will be implicitly applied during transformation to the target schema element.  See [NIEM-NDR] [Section 10.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.7), *The* "*Representation*" *pattern*.

##### Extends

UML::Property

##### Constraints

###### NDR3 [Rule 10-41] (REF,EXT). Name of element that ends in "Representation" is abstract

[Rule 10-41](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-41), Name of element that ends in Representation is abstract (REF, EXT): [Section 10.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.7), The Representation pattern

**[OCL] context** Representation **inv:**

(   
 (   
 not(self.namespace.oclIsUndefined())  
 and not(self.namespace.namespace.oclIsUndefined())  
 and self.namespace.namespace.stereotypedBy('InformationModel')  
 and not(self.name.oclIsUndefined())   
 and self.name.endsWith('Representation')  
 )   
 implies   
 self.stereotypedBy('Representation')  
)  
and  
(  
 self.stereotypedBy('Representation')  
 implies  
 self.type.oclIsUndefined()  
)

###### NDR3 [Rule 10-42] (REF,EXT). A substitution for a representation element declaration is a value for a type

[Rule 10-42](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-42), A substitution for a representation element declaration is a value for a type (REF, EXT): [Section 10](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10), Rules for NIEM modeling, by NIEM concept

[English]

Rule is definitional.

### <Stereotype> Restriction

##### Description

A Restriction Realization represents a relationship between two type definitions: the first is derived by restriction from the second. The two types must either both be NIEMType Classes or both be DataTypes. If the two types are Classes, then the attributes of the client class must be a subset of the attributes of the supplier class and omitted attributes must have a multiplicity lower bound of zero. if the two classes are DataTypes, then the client type is considered to have a value space that is a subset of that of the supplier, as may be further specified using a ValueRestriction stereotype on the client.  This relationship is implemented in XML Schema through the base attribute on the xsd:restriction element of the first type definition, the actual value of which resolves to the second type definition. If a type is a ValueRestriction the generalization owned by that type is implicitly an XSDRestriction. NIEM does not support the use of complex type restriction in reference schemas, because the use of restriction in a reference schema would reduce the ability for that schema to be reused. Restriction may be used in extension schemas.  Section 3.4 and 3.14 of [XML Schema Structures](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/) addresses the use of restriction in XML Schema.

##### Extends

UML::Realization

##### Constraints

###### XSDRestrictionComplexTypeComplexContent

If the general Classifier is a NIEMType that is not the client of a NIEMSimpleContent Realization, the specific Classifier must be a NIEMType that is not the client of a NIEMSimpleContent Realization.

**[OCL] context** Restriction **inv:**

(  
 self.base\_Realization.supplier.namespace->forAll(o|o.stereotypedBy('InformationModel'))   
 and  
 self.base\_Realization.supplier.clientDependency->select(d|d.stereotypedBy('NIEMSimpleContent'))->isEmpty()  
)   
implies  
 self.base\_Realization.client.clientDependency->select(d|d.stereotypedBy('NIEMSimpleContent'))->isEmpty()

###### XSDRestrictionComplexTypeSimpleContent

If the general Classifier is a NIEMType that is the client of a NIEMSimpleContent Realization, the specific Classifier must be a NIEMType that is the client of a NIEMSimpleContent Realization.

**[OCL] context** Restriction **inv:**

(  
 self.base\_Realization.supplier.namespace->forAll(o|o.stereotypedBy('InformationModel'))   
 and self.base\_Realization.supplier.clientDependency->select(d|d.stereotypedBy('XSDSimpleContent'))->notEmpty()  
)   
 implies(  
 self.base\_Realization.client.namespace->forAll(o|o.stereotypedBy('InformationModel'))   
 and self.base\_Realization.client.clientDependency->select(d|d.stereotypedBy('XSDSimpleContent'))->notEmpty()  
)

###### XSDRestrictionSimpleType

If the general Classifier is a DataType, the specific Classifier must be a DataType.

**[OCL] context** Restriction **inv:**

self.base\_Realization.supplier->forAll(o|o.oclIsKindOf(DataType)) implies  
 self.base\_Realization.client->forAll(o|o.oclIsKindOf(DataType))

### <Stereotype> Union

##### Description

A Union is a DataType whose value space is the union of one or more other DataTypes, which are the member types of the Union. The member types are specified using UnionOf Usage dependencies. A Union DataType is implemented in XML Schema as a union simple type definition. Each UnionOf dependency of which the Union is the client represents a relationship between two type definitions: the first is a union simple type definition whose member type definition is the second. This relationship is implemented in XML Schema through the memberTypes attribute on the xsd:union element of the union simple type definition, the actual value of which resolves to the second type definition. Section 3.14 of [XML Schema Structures](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/) addresses union simple type definitions in XML Schema.

##### Extends

UML::DataType

##### Constraints

###### NDR3 [Rule 11-8] (REF,EXT). Union member types defined by conformant schemas

[Rule 11-8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-8), Union member types defined by conformant schemas (REF, EXT): [Section 11.1.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1.2.2), Derivation by union

**[OCL] context** Union **inv:**

self.base\_DataType.clientDependency->select(d|d.stereotypedBy('UnionOf')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier)  
 ->forAll(t|(t.\_'package'.name='XMLPrimitiveTypes')or(t.\_'package'.appliedStereotype('Namespace').oclAsType(NIEM\_UML\_Profile::NIEM\_Common\_Profile::Namespace).isConformant))

###### NDR3 [Rule 9-17] (REF,EXT). No union member types of xs:ID

[Rule 9-17](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-17), No union member types of xs:ID (REF, EXT): [Section 9.1.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.2), Simple types prohibited as union member types

**[OCL] context** Union **inv:**

self.base\_DataType.clientDependency  
 ->select(d|d.stereotypedBy('UnionOf')).supplier  
 ->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier)  
 ->forAll(t|(t.name<>'ID')and(t.\_'package'.name<>'XMLPrimitiveTypes'))

###### NDR3 [Rule 9-18] (REF,EXT). No union member types of xs:IDREF

[Rule 9-18](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-18), No union member types of xs:IDREF (REF, EXT): [Section 9.1.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.2), Simple types prohibited as union member types

**[OCL] context** Union **inv:**

self.base\_DataType.clientDependency  
 ->select(d|d.stereotypedBy('UnionOf')).supplier  
 ->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier)  
->forAll(t|(t.name<>'IDREF')and(t.\_'package'.name<>'XMLPrimitiveTypes'))

###### NDR3 [Rule 9-19] (REF,EXT). No union member types of xs:IDREFS

[Rule 9-19](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-19), No union member types of xs:IDREFS (REF, EXT): [Section 9.1.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.2), Simple types prohibited as union member types

**[OCL] context** Union **inv:**

self.base\_DataType.clientDependency  
 ->select(d|d.stereotypedBy('UnionOf')).supplier  
 ->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier)  
->forAll(t|(t.name<>'IDREFS')and(t.\_'package'.name<>'XMLPrimitiveTypes'))

###### NDR3 [Rule 9-20] (REF,EXT). No union member types of xs:anySimpleType

[Rule 9-20](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-20), No union member types of xs:anySimpleType (REF, EXT): [Section 9.1.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.2), Simple types prohibited as union member types

**[OCL] context** Union **inv:**

self.base\_DataType.clientDependency  
 ->select(d|d.stereotypedBy('UnionOf')).supplier  
 ->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier)  
->forAll(t|(t.name<>'anySimpleType')and(t.\_'package'.name<>'XMLPrimitiveTypes'))

###### NDR3 [Rule 9-21] (REF,EXT). No union member types of xs:ENTITY

[Rule 9-21](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-21), No union member types of xs:ENTITY (REF, EXT): [Section 9.1.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.2), Simple types prohibited as union member types

**[OCL] context** Union **inv:**

self.base\_DataType.clientDependency  
 ->select(d|d.stereotypedBy('UnionOf')).supplier  
 ->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier)  
->forAll(t|(t.name<>'ENTITY')and(t.\_'package'.name<>'XMLPrimitiveTypes'))

###### NDR3 [Rule 9-22] (REF,EXT). No union member types of xs:ENTITIES

[Rule 9-22](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-22), No union member types of xs:ENTITIES (REF, EXT): [Section 9.1.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2.2), Simple types prohibited as union member types

**[OCL] context** Union **inv:**

self.base\_DataType.clientDependency  
 ->select(d|d.stereotypedBy('UnionOf')).supplier  
 ->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier)  
->forAll(t|(t.name<>'ENTITIES')and(t.\_'package'.name<>'XMLPrimitiveTypes'))

###### no generalizations

A Union shall not have any generalizations.

**[OCL] context** Union **inv:**

self.base\_DataType.generalization->isEmpty()

###### no owned attributes

A Union shall not have any ownedAttributes.

**[OCL] context** Union **inv:**

self.base\_DataType.attribute->isEmpty()

### <Stereotype> UnionOf

##### Description

The UnionOf stereotype is applied to a Usage dependency, the client of which must be a Union DataType and the supplier of which must be a DataType that represents a legal union member type. A UnionOf dependency specifies that the supplier DataType is a member type of the client Union.

##### Extends

UML::Usage

##### Constraints

###### client must be union

The client must be a union DataType

**[OCL] context** UnionOf **inv:**

self.base\_Usage.client->forAll(c|c.stereotypedBy('Union'))

###### supplier must be data type

The supplier must be a DataType that represents a legal union member type

**[OCL] context** UnionOf **inv:**

self.base\_Usage.supplier->forAll(s|s.oclIsKindOf(DataType))

### <Stereotype> ValueRestriction

##### Description

The ValueRestriction stereotype applies to a DataType (Enumeration or Primitive type) that is a specialization of a general DataType. It defines restrictions on which values of the general DataType that are allowed as values of the specialized DataType. A ValueRestriction DataType is implemented in XML Schema as a simple type definition that is a restriction of the simple type that implements the general DataType. The attributes of the ValueRestriction are implemented as restriction facets. ValueRestriction represents a NIEM type which is implemented in XML Schema as a simple type definition. The variety of the simple type definition may be union, list, or atomic. As the ValueRestriction stereotype is a specialization of DataType, it may be applied to Enumeration. In this case, the ValueRestriction represents a NIEM code type, which is implemented in XML Schema as a simple type definition that contains multiple xsd:enumeration facets.  Section 3.14 of [XML Schema Structures](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/) addresses simple type definitions in XML Schema. See [NIEM-NDR] [Section 9.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2), *Simple type definition*.

##### Extends

UML::DataType

##### Properties

###### fractionDigits : Integer [0..1]

A restriction on the value space of a numeric data type that places an upper limit on the arithmetic precision of decimal values. The value space is restricted to those values that can be represented lexically in decimal notation using at most fractionDigits to the right of the decimal point. fractionDigits is implemented in XML Schema as the value of the value attribute on the xsd:fractionDigits element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

###### length : Integer [0..1]

A restriction on the value space of a data type to values with a specific length, where the units of length depends on the base type being restricted. For String and URI values, the units are characters. For Binary values, the units are octets. For lists, the length is the number of items in the list. length is implemented in XML Schema as the value of the value attribute on the xsd:length element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

###### maxExclusive : String [0..1]

The exclusive upper bound of the value space for a data type with ordered values. The value of maxExclusive must be equal to some value in the value space of the base data type or to the maxExclusive restriction of the base type (if it has one). maxExclusive is implemented in XML Schema as the value of the value attribute on the xsd:maxExclusive element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

###### maxInclusive : String [0..1]

The inclusive upper bound of the value space for a data type with ordered values. The value of maxInclusive must be equal to some value in the value space of the base data type. maxInclusive is implemented in XML Schema as the value of the value attribute on the xsd:maxInclusive element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

###### maxLength : Integer [0..1]

A restriction on the value space of a data type to values with a specific maximum length, where the units of length depends on the base type being restricted. For String and URI values, the units are characters. For Binary values, the units are octets. For lists, the length is the number of items in the list. maxLength is implemented in XML Schema as the value of the value attribute on the xsd:maxLength element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

###### minExclusive : String [0..1]

The exclusive lower bound of the value space for a data type with ordered values. The value of minExclusive must be equal to some value in the value space of the base data type or to the minExclusive restriction of the base type (if it has one). minExclusive is implemented in XML Schema as the value of the value attribute on the xsd:minExclusive element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

###### minInclusive : String [0..1]

The inclusive lower bound of the value space for a data type with ordered values. The value of minInclusive must be equal to some value in the value space of the base data type. minInclusive is implemented in XML Schema as the value of the value attribute on the xsd:minInclusive element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

###### minLength : Integer [0..1]

A restriction on the value space of a data type to values with a specific minimum length, where the units of length depends on the base type being restricted. For String and URI values, the units are characters. For Binary values, the units are octets. For lists, the length is the number of items in the list. minLength is implemented in XML Schema as the value of the value attribute on the xsd:minLength element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

###### pattern : String [0..\*]

A constraint on the value space of a data type achieved by constraining the value space to those values represented by literals that match each member of a set of regular expressions. Each pattern must be a valid regular expression. pattern is implemented in XML Schema as the value of the value attribute on the xsd:pattern element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

###### totalDigits : Integer [0..1]

Restricts the magnitude and arithmetic precision of values in the value space of a numeric data type. The value space is restricted to those values that can be represented lexically using at most totalDigits digits in decimal notation or at most totalDigits digits for the coefficient, in scientific notation. totalDigits is implemented in XML Schema as the value of the value attribute on the xsd:totalDigits element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

##### Constraints

###### NDR3 [Rule 11-10] (REF,EXT). Code simple type has enumerations

[Rule 11-10](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-10), Code simple type has enumerations (REF, EXT): [Section 11.1.2.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1.2.3), Code simple types

**[OCL] context** ValueRestriction **inv:**

(  
 self.\_'package'.stereotypedBy('InformationModel')  
 and   
 not(self.name.oclIsUndefined())  
 and   
 self.name.endsWith('CodeSimpleType')  
)   
implies  
 (  
 (self.oclIsKindOf(Enumeration) and self.oclAsType(Enumeration).ownedLiteral->notEmpty())  
 or  
 self.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->forAll(s|s.name.endsWith('CodeSimpleType'))  
 or  
 (self.stereotypedBy('Union') and self.clientDependency->select(d|d.stereotypedBy('UnionOf')).supplier->exists(s|s.name.endsWith('CodeSimpleType'))   
 )

###### NDR3 [Rule 11-32] (REF,EXT). Standard opening phrase for complex type

[Rule 11-32](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-32), Standard opening phrase for complex type (REF, EXT): [Section 11.6.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.6.1.1), Data definition opening phrases

**[OCL] context** ValueRestriction **inv:**

(  
 not(self.stereotypedBy('LocalVocabulary'))   
 and  
 self.namespace.stereotypedBy('InformationModel')   
 )   
implies   
self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^a data type'))

###### NDR3 [Rule 11-9] (REF,EXT). Name of a code simple type has standard suffix

[Rule 11-9](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-9), Name of a code simple type has standard suffix (REF, EXT): [Section 11.1.2.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.1.2.3), Code simple types

**[OCL] context** ValueRestriction **inv:**

(  
 self.\_'package'.stereotypedBy('InformationModel')  
 and  
 (self.ownedLiteral->notEmpty() or self.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->forAll(s|s.name.endsWith('CodeSimpleType')))  
)   
implies  
self.name.endsWith('CodeSimpleType')

###### NDR3 [Rule 9-12] (REF,EXT). Simple type has data definition

[Rule 9-12](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-12), Simple type has data definition (REF, EXT): [Section 9.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2), Simple type definition

**[OCL] context** ValueRestriction **inv:**

(   
 not(self.namespace.oclIsUndefined())  
 and not(self.namespace.stereotypedBy('InformationModel'))  
)  
implies  
self.ownedComment->exists(c|not(c.\_'body'.oclIsUndefined())and(c.\_'body'<>''))

###### NDR3 [Rule10-17] (REF,EXT) Name of code type ends in CodeType

[Rule 10-17](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-17), Name of code type ends in CodeType (REF, EXT): [Section 10.2.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.4), Code types

**[OCL] context** ValueRestriction **inv:**

(  
 not(self.name.oclIsUndefined())  
 and  
 self.name.endsWith('CodeType')  
)   
implies  
self.general->union(self.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).oclAsType(Classifier))  
->forAll(c|c.name.endsWith('CodeType')or c.name.endsWith('CodeSimpleType'))

###### ValueRestriction Generalization

A valuerestriction that is not an enumeration or list may only generalize the

same metatype. A valuerestriction that is an enumeration may not generalize a list.

A valuerestriction that is a list may not generalize an enumeration.

###### ValueRestrictionGeneralization

A ValueRestriction DataType must be the special classifier in a single Generalization whose general classifier is also a DataType.

**[OCL] context** ValueRestriction **inv:**

self.base\_DataType.generalization.general->size()=1

## Profile : NIEM\_PIM\_Profile

### Overview

The NIEM PIM Profile comprises stereotypes that are used in NIEM PIMs but not NIEM PSMs. Further, the NIEM PIM Profile imports the NIEM Common Profile and, therefore, includes all the stereotypes and metaclasses covered by that profile. In addition, the UML metamodel subset covered by the NIEM PIM Profile also includes the metaclasses Association and AssociationClass, even though they are not specifically extended by any stereotypes in the profile.

### <Stereotype> Augments

##### Description

An Augments Generalization specifies that the special Class is an augmentation type that is restricted to augment instances of the general Class.

##### Extends

UML::Generalization

### <Stereotype> InformationModel

##### Description

The contents of an InformationModel Package provide a platform-independent perspective on the structure of information to be exchanged in NIEM messages. Such a model is always taken to represent a NIEM namespace, but it may also be given a default purpose as modeled, independent of the implementation of that namespace. This allows a modeler to identify the intended purposes (e.g., reference, subset, exchange, etc.) of various information models within a set, without having to create a complete MPD model for the set.

##### Generalization

[Namespace](#_a3b43d75feafe90b105d1a836eb3d6a2)

##### Properties

###### defaultPurpose : DefaultPurposeCode [0..1]

The default purpose for which an information model is intended. If an InformationModel Package is modeled as being included as an artifact in an MPD, then, unless otherwise specified, the purpose of the artifact is by default taken to be the schema purpose code corresponding to the value of the defaultPurpose attribute.

##### Constraints

###### NDR [Rule 6-55]

**[Rule 6-55] (REF)** Within the schema, the element xsd:complexContent MUST have as an immediate child the element xsd:extension.

**Rationale** NIEM does not support, as conformant, the use of complex type restriction. NIEM defines a language, in which specific content is allowed. It does not specify messages that forbid content. Such restrictions may be performed in nonconformant schemas or within constraint schemas or other artifacts of constraint.

Note that XML Schema requires use of the attribute base on xsd:extension.

Note also that the applicability allows for the use of restriction in subset schemas, extension schemas, exchange schemas, and constraint schemas.

**[OCL] context** InformationModel **inv:**

(self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference)  
implies  
self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).supplierDependency->select(d|d.stereotypedBy('Restriction'))->isEmpty()

###### NDR [Rule 6-57]

**[Rule 6-57] (EXT)** Within the schema, given an element xsd:complexContent with a child xsd:restriction owning an attribute base, the attribute base MUST have a value that resolves to the name of a complex type that is a NIEM-conformant component.

**[Rationale]** This ensures that a CCC defined through restriction has well-defined semantics.

**[OCL] context** InformationModel **inv:**

not(self.isConformant)  
implies  
   
self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).supplierDependency->select(d|d.stereotypedBy('Restriction'))->isEmpty()

###### NDR [Rule 9-26]

**[Rule 9-26] (REF, SUB, EXT)** Within the schema, any type definition which has a base type definition of a code type or which is transitively based on a code type SHALL have a name that uses the representation term qualifier Code.

**Rationale** This expands the use of the representation term qualifier Code to any type based on a code list.

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType   
 ->select(t|  
 t.clientDependency->select(d|d.stereotypedBy('XSDSimpleContent')).supplier->exists(s|s.oclIsKindOf(Enumeration))  
 or (  
 t.oclIsKindOf(Classifier)   
 and t.oclAsType(Classifier).allParents().clientDependency->select(d|d.stereotypedBy('XSDSimpleContent')).supplier->exists(s|s.oclIsKindOf(Enumeration))  
 )  
 )  
 ->forAll(enumerationDerivedType|enumerationDerivedType.name.match('.\*Code.\*'))

###### NDR [Rule 9-32]

**[Rule 9-32] (REF, SUB, EXT)** Within the schema, the name of an association element SHALL use the representation term qualifier Association.

**Rationale** Using the qualifier Association immediately identifies an element as representing an association.

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).attribute->select(a|not(a.type.oclIsUndefined()))  
 ->forAll(a|a.type.stereotypedBy('AssociationType') implies a.name.match('.\*Association.\*'))

###### NDR [Rule 9-33]

**[Rule 9-33] (REF, SUB, EXT)** Within the schema, the name of an augmentation element SHALL use the representation term Augmentation.

**Rationale** Using the qualifier Augmentation immediately identifies an element as representing an augmentation.

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).attribute->select(a|not(a.type.oclIsUndefined()))  
 ->forAll(a|a.type.stereotypedBy('AugmentationType') implies a.name.match('.\*Augmentation.\*'))

###### NDR [Rule 9-34]

**[Rule 9-34] (REF, SUB, EXT)** Within the schema, the name of a metadata element SHALL use the representation term Metadata.

**Rationale** Using the qualifier Metadata immediately identifies an element as representing metadata.

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).attribute  
->forAll(a|(not(a.type.oclIsUndefined()) and a.type.stereotypedBy('MetadataType')) implies a.name.match('.\*Metadata.\*'))

###### NDR3 [Rule 10-1] (REF,EXT). Complex type has a category

[Rule 10-1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-1), Complex type has a category (REF, EXT): [Section 10.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.1), Categories of NIEM type definitions

[English]

The constraint is satisfied during provisioning, which produce one of the NDR defined complex type categories based on explicit or implicit model specifications.

###### NDR3 [Rule 10-2] (REF,EXT). Object type with complex content is derived from object type

[Rule 10-2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-2), Object type with complex content is derived from object type (REF, EXT): [Section 10.2.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.1.1), Object types with complex content

[English]

The constraint is satisfied during provisioning, which produce derivation of each Object Type from another Object Type or, if not modeled explicitly, from structures:ObjectType.

###### NDR3 [Rule 10-43] (REF,EXT). Schema component name composed of English words

[Rule 10-43](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-43), Schema component name composed of English words (REF, EXT): [Section 10.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8), Naming rules

[English]

This rule is not readily computational.

###### NDR3 [Rule 10-44] (REF,EXT). Schema component names have only specific characters

[Rule 10-44](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-44), Schema component names have only specific characters (REF, EXT): [Section 10.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8), Naming rules

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType ->select(t|not(t.stereotypedBy('PropertyHolder')))  
 ->forAll(schemaComponent|schemaComponent.name.match('[\\w|\\-]\*'))   
and   
self.base\_Package.ownedType ->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).attribute  
 ->forAll(schemaComponent|schemaComponent.name.match('[\\w|\\-]\*'))

###### NDR3 [Rule 10-45] (REF,EXT). Hyphen in component name is a separator

[Rule 10-45](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-45), Hyphen in component name is a separator (REF, EXT): [Section 10.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8), Naming rules

[English]

Rule is definitional.

###### NDR3 [Rule 10-46](REF,EXT). Names use camel case

[Rule 10-46](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-46), Names use camel case (REF, EXT): [Section 10.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.1), Character case

[English]

Rule is not reliably computational.

###### NDR3 [Rule 10-48] (REF,EXT). Name of schema component other than attribute begins with upper case letter

[Rule 10-48](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-48), Name of schema component other than attribute begins with upper case letter (REF, EXT): [Section 10.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.1), Character case

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType  
 ->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier )  
 ->forAll(c|  
 c.oclAsType(NamedElement)->asSet()  
 ->union(c.attribute  
 ->select(a|   
 not(a.stereotypedBy('XSDProperty'))  
 or   
 not(a.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)  
 )  
 .oclAsType(NamedElement)->asSet()  
 )  
 ->forAll(n|not(n.name.oclIsUndefined()or (n.name='')) implies (n.name.firstToUpper()=n.name))  
)

###### NDR3 [Rule 10-49] (REF,EXT). Names use common abbreviations

[Rule 10-49](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-49), Names use common abbreviations (REF, EXT): [Section 10.8.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.2), Use of acronyms and abbreviations

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType  
 ->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)  
 ->forAll(c|  
 c.oclAsType(NamedElement)->asSequence()  
 ->union(c.attribute.oclAsType(NamedElement))  
 ->forAll(n|not(n.name.oclIsUndefined()or (n.name=''))   
 implies   
 (n.name.match('.\*Identifier.\*')or n.name.match('.\*UniformResourceIdentifier.\*'))  
 )  
)

###### NDR3 [Rule 10-4] (REF,EXT). Only object type has RoleOf element

[Rule 10-4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-4), Only object type has RoleOf element (REF, EXT): [Section 10.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.2), Role types and roles

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)  
 ->select(t|t.stereotypedBy('MetadataType')or t.stereotypedBy('AssociationType')or t.stereotypedBy('AugmentationType') or t.oclIsKindOf(AssociationClass)).attribute  
 ->forAll(a|not(a.stereotypedBy('RoleOf') or a.name.startsWith('RoleOf')))

###### NDR3 [Rule 10-50] (REF,EXT). Local term declaration is local to its schema document

[Rule 10-50](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-50), Local term declaration is local to its schema document (REF, EXT): [Section 10.8.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.2.1), Use of Acronyms, Initialisms, Abbreviations, and Jargon

[English]

Rule is definitional.

###### NDR3 [Rule 10-51] (REF,EXT). Local terminology interpretation

[Rule 10-51](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-51), Local terminology interpretation (REF, EXT): [Section 10.8.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.2.1), Use of Acronyms, Initialisms, Abbreviations, and Jargon

[English]

Rule is definitional.

###### NDR3 [Rule 10-52] (REF,EXT). Singular form is preferred in name

[Rule 10-52](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-52), Singular form is preferred in name (REF, EXT): [Section 10.8.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.3), Word forms

[English]

Rule is definitional.

###### NDR3 [Rule 10-53] (REF,EXT). Present tense is preferred in name

[Rule 10-53](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-53), Present tense is preferred in name (REF, EXT): [Section 10.8.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.3), Word forms

[English]

Rule is definitional.

###### NDR3 [Rule 10-54] (REF,EXT). Name does not have nonessential words

[Rule 10-54](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-54), Name does not have nonessential words (REF, EXT): [Section 10.8.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.3), Word forms

[English]

Rule is definitional.

###### NDR3 [Rule 10-55] (REF,EXT). Component name follows pattern

[Rule 10-55](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-55), Component name follows pattern (REF, EXT): [Section 10.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8), Naming rules

[English]

Rule is definitional.

###### NDR3 [Rule 10-56] (REF,EXT). Object-class term identifies concrete category

[Rule 10-56](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-56), Object-class term identifies concrete category (REF, EXT): [Section 10.8.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.4), Object-class term

[English]

Rule is definitional.

###### NDR3 [Rule 10-57] (REF,EXT). Property term describes characteristic or subpart

[Rule 10-57](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-57), Property term describes characteristic or subpart (REF, EXT): [Section 10.8.5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.5), Property term

[English]

Rule is definitional.

###### NDR3 [Rule 10-58] (REF,EXT). Name may have multiple qualifier terms

[Rule 10-58](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-58), Name may have multiple qualifier terms (REF, EXT): [Section 10.8.6](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.6), Qualifier terms

[English]

Rule is definitional.

###### NDR3 [Rule 10-59] (REF,EXT). Name has minimum necessary number of qualifier terms

[Rule 10-59](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-59), Name has minimum necessary number of qualifier terms (REF, EXT): [Section 10.8.6](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.6), Qualifier terms

[English]

Rule is definitional.

###### NDR3 [Rule 10-5] (REF,EXT,INS). RoleOf elements indicate the base types of a role type

[Rule 10-5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-5), RoleOf elements indicate the base types of a role type (REF, EXT, INS): [Section 10.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.2), Role types and roles

[English]

This rule is definitional.

###### NDR3 [Rule 10-60] (REF,EXT). Order of qualifies is not significant

[Rule 10-60](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-60), Order of qualifies is not significant (REF, EXT): [Section 10.8.6](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.6), Qualifier terms

[English]

Rule is definitional.

###### NDR3 [Rule 10-61] (REF,EXT). Redundant term in name is omitted

[Rule 10-61](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-61), Redundant term in name is omitted (REF, EXT): [Section 10.8.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.7), Representation terms

[English]

The constraint can not be expressed easily in OCL.

###### NDR3 [Rule 10-65](REF,EXT). Machine-readable annotations are valid

[Rule 10-65](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-65), Machine-readable annotations are valid (REF, EXT): [Section 10.9](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9), Machine-readable annotations

[English]

The constraint is realized through provisioning:  
there are no NIEM-UML constructs related specifically to machine-readable annotations;  
the production of machine-readable annotations is based on the mapping of specific NDR rules to target schema annotations.

###### NDR3 [Rule 10-67] (REF,EXT). Deprecated annotates schema component

[Rule 10-67](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-67), Deprecated annotates schema component (REF, EXT): [Section 10.9.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1.1), Deprecation

[English]

The constraint is realized through provisioning:  
A NamedElement with applied Stereotype Deprecated will create the appinfo:deprecated on the target schema component.

###### NDR3 [Rule 10-68] (REF,EXT). External import indicator annotates import

[Rule 10-68](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-68), External import indicator annotates import (REF, EXT): [Section 10.9.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.9.1), The NIEM appinfo namespace

[English]

The constraint is realized through provisioning:  
A provisioned xs:import will own an appinfo:externalImportIndicator if the import InformationModel has a defaultPurpose of external.

###### NDR3 [Rule 10-6] (INS). Instance of RoleOf element indicates a role object

[Rule 10-6](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-6), Instance of RoleOf element indicates a role object (INS): [Section 10.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.2), Role types and roles

[English]

This rule is definitional.

###### NDR3 [Rule 10-76] (REF,EXT,INS). Use structures as specified

[Rule 10-76](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-76), Use structures as specified (REF, EXT, INS): [Section 10.10](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.10), NIEM structural facilities

[English]

The constraint is realized through provisioning:  
The structures namespace is not part of the NIEM-UML model, all usages of the namespace are provisioned according to the NDR rules governing that namespace.

###### NDR3 [Rule 11-24] (REF,EXT). Schema uses only known attribute groups

[Rule 11-24](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-24), Schema uses only known attribute groups (REF, EXT): [Section 11.3.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.3.3.1), Attribute group use

[English]

Expression of this constraint as OCL has been deferred.

###### NDR3 [Rule 11-25] (REF,EXT). Data definition does not introduce ambiguity

[Rule 11-25](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-25), Data definition does not introduce ambiguity (REF, EXT): [Section 11.6.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.6.1), Human-readable documentation

[English]

Constraint is non-computable.

###### NDR3 [Rule 11-26] (REF,EXT). Object class has only one meaning

[Rule 11-26](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-26), Object class has only one meaning (REF, EXT): [Section 11.6.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.6.1), Human-readable documentation

[English]

Constraint is non-computable.

###### NDR3 [Rule 11-27] (REF,EXT). Data definition of a part does not redefine the whole

[Rule 11-27](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-27), Data definition of a part does not redefine the whole (REF, EXT): [Section 11.6.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.6.1), Human-readable documentation

[English]

Constraint is non-computable.

###### NDR3 [Rule 11-28] (REF,EXT). Do not leak representation into data definition

[Rule 11-28](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-28), Do not leak representation into data definition (REF, EXT): [Section 11.6.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.6.1), Human-readable documentation

[English]

Constraint is non-computable.

###### NDR3 [Rule 11-29] (REF,EXT). Data definition follows 11179-4 requirements

[Rule 11-29](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-29), Data definition follows 11179-4 requirements (REF, EXT): [Section 11.6.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.6.1), Human-readable documentation

[English]

Constraint is non-computable.

###### NDR3 [Rule 11-30] (REF,EXT). Data definition follows 11179-4 recommendations

[Rule 11-30](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-30), Data definition follows 11179-4 recommendations (REF, EXT): [Section 11.6.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.6.1), Human-readable documentation

[English]

Constraint is non-computable.

###### NDR3 [Rule 11-34] (REF,EXT). Same namespace means same components

[Rule 11-34](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-34), Same namespace means same components (REF, EXT): [Section 11.7.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.7.1), xs:schema document element restrictions

[English]

Constraint expression as OCL is deferred.

###### NDR3 [Rule 11-35] (REF,EXT). Different version means different view

[Rule 11-35](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-35), Different version means different view (REF, EXT): [Section 11.7.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.7.1), xs:schema document element restrictions

[English]

Rule is definitional.

###### NDR3 [Rule 11-36] (SET). Reference schema imports reference schema

[Rule 11-36](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-36), Reference schema imports reference schema (SET): [Section 11.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.8), Schema assembly

**[OCL] context** InformationModel **inv:**

(  
 (self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference)  
 or  
 (self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::subset)  
)   
implies  
self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general.namespace.oclAsType(NamedElement)->asSet()  
 ->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).namespace.oclAsType(NamedElement)->asSet())  
 ->union(c.attribute->select(a|not(a.type.oclIsUndefined())).type.namespace.oclAsType(NamedElement)->asSet())  
 ->union(c.attribute.clientDependency->select(d|d.stereotypedBy('References')).supplier->select(s|s.oclIsKindOf(Property)).namespace.namespace.oclAsType(NamedElement)->asSet())  
 .appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel).defaultPurpose->forAll(p|(p=DefaultPurposeCode::reference)or(p=DefaultPurposeCode::subset))  
)

###### NDR3 [Rule 11-37] (SET). Extension schema document imports reference or extension schema

[Rule 11-37](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-37), Extension schema document imports reference or extension schema (SET): [Section 11.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.8), Schema assembly

**[OCL] context** InformationModel **inv:**

(self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::extension)  
implies  
self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(c|  
 c.general.namespace.oclAsType(NamedElement)->asSet()  
 ->union(c.clientDependency->select(d|d.stereotypedBy('Restriction')).supplier->select(s|s.oclIsKindOf(Classifier)).namespace.oclAsType(NamedElement)->asSet())  
 ->union(c.attribute->select(a|not(a.type.oclIsUndefined())).type.namespace.oclAsType(NamedElement)->asSet())  
 ->union(c.attribute.clientDependency->select(d|d.stereotypedBy('References')).supplier->select(s|s.oclIsKindOf(Property)).namespace.namespace.oclAsType(NamedElement)->asSet())  
 ->select(p|p.stereotypedBy('InformationModel'))  
 .appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel).defaultPurpose->forAll(p|  
 (p=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference)  
 or(p=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::subset)  
 or(p=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::extension))  
)

###### NDR3 [Rule 11-38] (REF,EXT). Structures imported as conformant

[Rule 11-38](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-38), Structures imported as conformant (REF, EXT): [Section 11.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.8.1), Supporting namespaces are imported as conformant

[English]

Constraint realized by provisioning;  
there is no explicit modeling of xs:import in NIEM-UML; xs:import is produced as required, according to this and other NDR rules

###### NDR3 [Rule 11-39] (REF,EXT). XML namespace imported as conformant

[Rule 11-39](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-39), XML namespace imported as conformant (REF, EXT): [Section 11.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.8.1), Supporting namespaces are imported as conformant

[English]

Constraint realized by provisioning;  
there is no explicit modeling of xs:import in NIEM-UML; xs:import is produced as required, according to this and other NDR rules

###### NDR3 [Rule 11-40] (SET). Each namespace may have only a single root schema in a schema set

[Rule 11-40](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-40), Each namespace may have only a single root schema in a schema set (SET): [Section 11.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.8), Schema assembly

[English]

Expressing Constraint in OCL has been deferred

###### NDR3 [Rule 11-41] (REF,EXT). Consistently marked namespace imports

[Rule 11-41](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-41), Consistently marked namespace imports (REF, EXT): [Section 11.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.8), Schema assembly

[English]

Constraint ensured by provisioning:  
xs:import is not in the NIEM-UML Model, it is created during provisioning, which consistently constructs the externalImportIndicator based on the tag values in the referenced InformationModel.

###### NDR3 [Rule 12-10] (INS). Values of structures:metadata refer to values of structures:id

[Rule 12-10](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-10), Values of structures:metadata refer to values of structures:id (INS): [Section 12.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.3), Instance metadata

[English]

Constraint is realized during provisioning of XML instance documents.

###### NDR3 [Rule 12-11] (INS). Value of structures:relationshipMetadata refers to value of structures:id

[Rule 12-11](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-11), Value of structures:relationshipMetadata refers to value of structures:id (INS): [Section 12.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.3), Instance metadata

[English]

Constraint realized during provisioning of XML instance documents.

###### NDR3 [Rule 12-12] (INS). structures:metadata and structures:relationshipMetadata refer to metadata elements

[Rule 12-12](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-12), structures:metadata and structures:relationshipMetadata refer to metadata elements (INS): [Section 12.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.3), Instance metadata

[English]

Constraint realized by provisioning of the XML Instance Documents.

###### NDR3 [Rule 12-13] (INS). Attribute structures:metadata references metadata element

[Rule 12-13](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-13), Attribute structures:metadata references metadata element (INS): [Section 12.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.3), Instance metadata

[English]

Constraint realized during provisioning of XML instance documents.

###### NDR3 [Rule 12-14] (INS). Attribute structures:relationshipMetadata references metadata element

[Rule 12-14](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-14), Attribute structures:relationshipMetadata references metadata element (INS): [Section 12.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.3), Instance metadata

[English]

Constraint realized by provisioning of XML Instance Documents.

###### NDR3 [Rule 12-15] (INS). Metadata is applicable to element

Rule 12-15, Metadata is applicable to element (INS): Section 12.3, Instance metadata

[English]

Constraint realized when provisioning XML Instance Document.

###### NDR3 [Rule 12-1] (INS). Instance must be schema-valid

[Rule 12-1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-1), Instance must be schema-valid (INS): [Section 12](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12), XML instance document rules

[English]

Constraint can not be easily expressed in OCL, the constraint must be enforced by an XML Schema Document Validation tool.

###### NDR3 [Rule 12-2] (INS). Element with structures:ref does not have content

[Rule 12-2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-2), Element with structures:ref does not have content (INS): [Section 12.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.2), Reference elements

[English]

Constraint is realized during provisioning of instance documents, if any. Provisioning of an element with @structures:ref attribute will not have element content.

###### NDR3 [Rule 12-3] (INS). Attribute structures:ref must reference structures:id

[Rule 12-3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-3), Attribute structures:ref must reference structures:id (INS): [Section 12.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.2), Reference elements

[English]

Constraint is realized during provisioning of instance documents, if any. Any @structures:ref will reference an element with the same value in an @structures:id.

###### NDR3 [Rule 12-3] (INS). Attribute structures:ref must reference structures:id2

[Rule 12-3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-3), Attribute structures:ref must reference structures:id (INS): [Section 12.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.2), Reference elements

[English]

Constraint is realized during provisioning of instance documents, if any. Any @structures:ref will reference an element with the same value in an @structures:id.

###### NDR3 [Rule 12-4] (INS). Linked elements have same validation root

[Rule 12-4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-4), Linked elements have same validation root (INS): [Section 12.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.2), Reference elements

[English]

Constraint is realized during provisioning of instance documents, if any.

###### NDR3 [Rule 12-5] (INS). Attribute structures:ref references element of correct type

[Rule 12-5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-5), Attribute structures:ref references element of correct type (INS): [Section 12.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.2), Reference elements

[English]

Constraint is realized during provisioning of instance documents, if any.

###### NDR3 [Rule 12-6] (INS). Reference and content elements have the same meaning

[Rule 12-6](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-6), Reference and content elements have the same meaning (INS): [Section 12.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.2.1), Reference and content elements have same meaning

[English]

Rule is definitional.

###### NDR3 [Rule 12-7] (INS). Empty content has no meaning

[Rule 12-7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-7), Empty content has no meaning (INS): [Section 12](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12), XML instance document rules

[English]

Rule is definitional.

###### NDR3 [Rule 12-8] (INS). Metadata applies to referring entity

[Rule 12-8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-8), Metadata applies to referring entity (INS): [Section 12.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.3), Instance metadata

[English]

Rule is definitional.

###### NDR3 [Rule 12-9] (INS). Referent of structures:relationshipMetadata annotates relationship

[Rule 12-9](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_12-9), Referent of structures:relationshipMetadata annotates relationship (INS): [Section 12.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_12.3), Instance metadata

[English]

Rule is definitional.

###### NDR3 [Rule 4-1] (SET) Schema marked as reference schema document must conform

[Rule 4-1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_4-1), Schema marked as reference schema document must conform (SET): [Section 4.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_4.1), Conformance targets defined

[English]

This constraint realized by the aggregate of constraints targeting REF schema documents.

###### NDR3 [Rule 4-2] (SET) Schema marked as extension schema document must conform

[Rule 4-2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_4-2), Schema marked as extension schema document must conform (SET): [Section 4.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_4.1), Conformance targets defined

[English]

This constraint realized by the aggregate of constraints targeting EXT schema documents.

###### NDR3 [Rule 4-3] (REF,EXT) Schema is CTAS-conformant

[Rule 4-3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_4-3), Schema is CTAS-conformant (REF, EXT): [Section 4.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_4.3), Conformance target identifiers

[English]

This constraint realized by the aggregate of constraints targeting REF and EXT schema documents.

###### NDR3 [Rule 4-4] (REF,EXT). Document element has attribute ct:conformanceTargets

[Rule 4-4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_4-4), Document element has attribute ct:conformanceTargets (REF, EXT): [Section 4.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_4.3), Conformance target identifiers

[English]

This constraint realized during provisioning of the schema associated with the InformationModel

###### NDR3 [Rule 4-5] (REF). Schema claims reference schema conformance target

[Rule 4-5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_4-5), Schema claims reference schema conformance target (REF): [Section 4.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_4.3), Conformance target identifiers

[English]

This constraint realized during provisioning of the schema associated with the InformationModel

###### NDR3 [Rule 4-6] (EXT). Schema claims extension conformance target

[Rule 4-6](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_4-6), Schema claims extension conformance target (EXT): [Section 4.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_4.3), Conformance target identifiers

[English]

This constraint realized during provisioning of the schema associated with the InformationModel

###### NDR3 [Rule 7-1] (REF,EXT,INS). Document is an XML document

[Rule 7-1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_7-1), Document is an XML document (REF, EXT, INS): [Section 7.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_7.1), Conformance to XML

[English]

This constraint realized during provisioning of the schema associated with the InformationModel

###### NDR3 [Rule 9-11] (REF). No simple type disallowed derivation

[Rule 9-11](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-11), No simple type disallowed derivation (REF): [Section 9.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.2), Simple type definition

**[OCL] context** InformationModel **inv:**

((self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference)or(self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::subset))  
implies  
self.base\_Package.ownedType->select(t|t.oclIsKindOf(DataType)).oclAsType(DataType)->forAll(dt|not(dt.isFinalSpecialization))

###### NDR3 [Rule 9-29] (REF). Complex content uses extension

[Rule 9-29](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-29), Complex content uses extension (REF): [Section 9.1.3.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3.2), Complex content

**[OCL] context** InformationModel **inv:**

((self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference)or(self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::subset))  
implies  
self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).clientDependency  
 ->select(d|d.stereotypedBy('Restriction'))->isEmpty()

###### NDR3 [Rule 9-30] (REF,EXT). Base type of complex type with complex content must have complex content

[Rule 9-30](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-30), Base type of complex type with complex content must have complex content (REF, EXT): [Section 9.1.3.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3.2.1), Base type of complex type with complex content has complex content

[English]

Provisioning to target schemas ensures the base type of Complex types with complex content will have complex content.

###### NDR3 [Rule 9-31] (SET). Base type of complex type with complex content must have complex content

[Rule 9-31](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-31), Base type of complex type with complex content must have complex content (SET): [Section 9.1.3.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3.2.1), Base type of complex type with complex content has complex content

[English]

Provisioning to target schemas ensures the base type of Complex types with complex content will have complex content.

###### NDR3 [Rule 9-32] (REF). Simple content uses extension

[Rule 9-32](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-32), Simple content uses extension (REF): [Section 9.1.3.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3.3), Simple content

**[OCL] context** InformationModel **inv:**

((self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference)or(self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::subset))  
implies  
self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).clientDependency  
 ->select(d|d.stereotypedBy('Restriction'))->isEmpty()

###### NDR3 [Rule 9-33] (REF). No complex type disallowed substitutions

[Rule 9-33](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-33), No complex type disallowed substitutions (REF): [Section 9.1.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3), Complex type definition

[English]

The concept of disallowed substitutions is currently not supported by the NIEM UML Profile. Currently, there will be no "block" provisioned for a complex type.

###### NDR3 [Rule 9-34] (REF). No complex type disallowed derivation

[Rule 9-34](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-34), No complex type disallowed derivation (REF): [Section 9.1.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3), Complex type definition

**[OCL] context** InformationModel **inv:**

((self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference)or(self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::subset))  
implies  
self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)->forAll(dt|not(dt.isFinalSpecialization))

###### NDR3 [Rule 9-35] (REF,EXT). Element declaration is top-level

[Rule 9-35](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-35), Element declaration is top-level (REF, EXT): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

[English]

Constraint is enforced during provisioning, top level elements are always created and referenced by non top level elements.

###### NDR3 [Rule 9-36] (REF,EXT). Element declaration has data definition

[Rule 9-36](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-36), Element declaration has data definition (REF, EXT): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).attribute  
 ->forAll(a|a.ownedComment->exists(c|not(c.\_'body'.oclIsUndefined()) and (c.\_'body'<>'')))

###### NDR3 [Rule 9-39] (REF,EXT). Element type not in the XML Schema namespace

[Rule 9-39](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-39), Element type not in the XML Schema namespace (REF, EXT): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

**[OCL] context** InformationModel **inv:**

self.targetNamespace<>'http://www.w3.org/2001/XMLSchema'

###### NDR3 [Rule 9-40] (REF,EXT). Element type not in the XML namespace

[Rule 9-40](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-40), Element type not in the XML namespace (REF, EXT): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).attribute.type->forAll(t| t.\_'package'.name<>'xml')

###### NDR3 [Rule 9-41] (REF,EXT). Element type is not simple type

[Rule 9-41](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-41), Element type is not simple type (REF, EXT): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

[English]

OCL representation of this constraint is deferred.

###### NDR3 [Rule 9-42] (REF). No element disallowed substitutions

[Rule 9-42](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-42), No element disallowed substitutions (REF): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

[English]

The concept of disallowed substitutions (@block) is currently not supported by NIEM-UML. There will be no provisioning of the @block attribute.

###### NDR3 [Rule 9-47] (REF,EXT). Attribute declaration is top-level

[Rule 9-47](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-47), Attribute declaration is top-level (REF, EXT): [Section 9.2.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3), Attribute declaration

[English]

Constraint is enforced during provisioning, top level attributes are always created and referenced by non top level elements.

###### NDR3 [Rule 9-48] (REF,EXT). Attribute declaration has data definition

[Rule 9-48](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-48), Attribute declaration has data definition (REF, EXT): [Section 9.2.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3), Attribute declaration

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedType->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier).attribute  
 ->forAll(a|a.ownedComment->exists(c|not(c.\_'body'.oclIsUndefined()) and (c.\_'body'<>'')))

###### NDR3 [Rule 9-49] (REF,EXT). Attribute declaration has type

[Rule 9-49](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-49), Attribute declaration has type (REF, EXT): [Section 9.2.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3), Attribute declaration

[English]

Specification as OCL Constraint is deferred.

###### NDR3 [Rule 9-61] (REF). xs:sequence must be child of xs:extension

[Rule 9-61](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-61), xs:sequence must be child of xs:extension (REF): [Section 9.3.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.1.1), Sequence

[English]

Constraint enforced by provisioning, an xs:sequence is always produced as a child of an xs:extension.

###### NDR3 [Rule 9-62] (EXT). xs:sequence must be child of xs:extension or xs:restriction

[Rule 9-62](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-62), xs:sequence must be child of xs:extension or xs:restriction (EXT): [Section 9.3.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.1.1), Sequence

[English]

Constraint enforced by provisioning, an xs:sequence is always produced as a child of an xs:extension or xs:restriction.

###### NDR3 [Rule 9-63] (REF). No xs:choice

[Rule 9-63](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-63), No xs:choice (REF): [Section 9.3.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.1.2), Choice

**[OCL] context** InformationModel **inv:**

((self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::subset)or(self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference))  
implies  
self.base\_Package.ownedType  
->select(t|t.oclIsKindOf(Classifier)).oclAsType(Classifier)   
->union(self.base\_Package.ownedType->select(t|t.oclIsKindOf(Class)).oclAsType(Class).nestedClassifier)  
->select(t|t.stereotypedBy('Choice'))->isEmpty()

###### NDR3 [Rule 9-64] (EXT). xs:choice must be child of xs:sequence

[Rule 9-64](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-64), xs:choice must be child of xs:sequence (EXT): [Section 9.3.1.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.1.2), Choice

[English]

Constraint enforced by provisioning, an xs:choice is always produced as a child of an xs:sequence.

###### NDR3 [Rule 9-65] (REF,EXT). Sequence has minimum cardinality 1

[Rule 9-65](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-65), Sequence has minimum cardinality 1 (REF, EXT): [Section 9.3.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.2.1), Sequence cardinality

[English]

Constraint enforced by provisioning, an xs:sequence is always produced with @minOccurs=1.

###### NDR3 [Rule 9-66] (REF,EXT). Sequence has maximum cardinality 1

[Rule 9-66](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-66), Sequence has maximum cardinality 1 (REF, EXT): [Section 9.3.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.2.1), Sequence cardinality

[English]

Constraint enforced by provisioning, an xs:sequence is always produced with @maxOccurs=1.

###### NDR3 [Rule 9-67] (EXT). Choice has minimum cardinality 1

[Rule 9-67](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-67), Choice has minimum cardinality 1 (EXT): [Section 9.3.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.2.2), Choice cardinality

[English]

Constraint enforced by provisioning, an xs:choice is always produced with @minOccurs=1.

###### NDR3 [Rule 9-68] (EXT). Choice has maximum cardinality 1

[Rule 9-68](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-68), Choice has maximum cardinality 1 (EXT): [Section 9.3.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.2.2), Choice cardinality

[English]

Constraint enforced by provisioning, an xs:choice is always produced with @maxOccurs=1.

###### NDR3 [Rule 9-69] (REF). No use of xs:any

[Rule 9-69](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-69), No use of xs:any (REF): [Section 9.3.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.4), Wildcard

**[OCL] context** InformationModel **inv:**

((self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference)or(self.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::subset))  
implies  
self.base\_Package.ownedType->forAll(dt|not(dt.stereotypedBy('XSDAnyProperty')))

###### NDR3 [Rule 9-70] (REF). No use of xs:anyAttribute

[Rule 9-70](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-70), No use of xs:anyAttribute (REF): [Section 9.3.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.3.4), Wildcard

[English]

Specification of this constraint as OCL has been deferred.

###### NDR3 [Rule 9-71] (REF,EXT). No use of xs:unique

[Rule 9-71](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-71), No use of xs:unique (REF, EXT): [Section 9.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.4), Identity-constraint definition components

[English]

Constraint enforced by provisioning, an xs:unique can not be modeled nor is it produced in a target schema.

###### NDR3 [Rule 9-72] (REF,EXT). No use of xs:key

[Rule 9-72](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-72), No use of xs:key (REF, EXT): [Section 9.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.4), Identity-constraint definition components

[English]

Constraint enforced by provisioning, an xs:key can not be modeled nor is it produced in a target schema.

###### NDR3 [Rule 9-73] (REF,EXT). No use of xs:keyref

[Rule 9-73](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-73), No use of xs:keyref (REF, EXT): [Section 9.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.4), Identity-constraint definition components

[English]

Constraint enforced by provisioning, an xs:keyref can not be modeled nor is it produced in a target schema.

###### NDR3 [Rule 9-74] (REF,EXT). No use of xs:group

[Rule 9-74](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-74), No use of xs:group (REF, EXT): [Section 9.5.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.5.1), Model group definition

[English]

Constraint enforced by provisioning, an xs:group can not be modeled nor is it produced in a target schema.

###### NDR3 [Rule 9-75] (REF,EXT). No definition of attribute groups

[Rule 9-75](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-75), No definition of attribute groups (REF, EXT): [Section 9.5.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.5.2), Attribute group definition

[English]

Constraint enforced by provisioning, an xs:attributeGroup can not be modeled nor is it produced in a target schema.

###### NDR3 [Rule 9-76] (REF,EXT). Comment is not recommended

[Rule 9-76](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-76), Comment is not recommended (REF, EXT): [Section 9.6](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.6), Annotation components

[English]

Constraint enforced by provisioning, an XML comment can not be modeled nor is it produced in a target schema.

###### NDR3 [Rule 9-77] (REF,EXT). Documentation element has no element children

[Rule 9-77](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-77), Documentation element has no element children (REF, EXT): [Section 9.6](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.6), Annotation components

[English]

Constraint enforced by provisioning, the xs:documentation is populated by a UML Comment body, which is a String (possibly escaped to ensure no nested xml elements are present).

###### NDR3 [Rule 9-78] (REF,EXT). xs:appinfo children are comments, elements, or whitespace

[Rule 9-78](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-78), xs:appinfo children are comments, elements, or whitespace (REF, EXT): [Section 9.6.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.6.1), Application information annotation

[English]

Constraint enforced by provisioning; the xs:appinfo is not directly modeled, and is provisioned in accordance with NDR-specified rules associated with specific NIEM concepts. Thus, an XML element is the child of an xs:appinfo.

###### NDR3 [Rule 9-79] (REF,EXT). Appinfo child elements have namespaces

[Rule 9-79](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-79), Appinfo child elements have namespaces (REF, EXT): [Section 9.6.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.6.1), Application information annotation

[English]

Constraint enforced by provisioning; the xs:appinfo is not directly modeled, and is provisioned in accordance with NDR-specified rules associated with specific NIEM concepts. Thus, an XML element is the child of an xs:appinfo and will have a namespace name.

###### NDR3 [Rule 9-80] (REF,EXT). Appinfo descendants are not XML Schema elements

[Rule 9-80](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-80), Appinfo descendants are not XML Schema elements (REF, EXT): [Section 9.6.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.6.1), Application information annotation

[English]

Constraint enforced by provisioning; the xs:appinfo is not directly modeled, and is provisioned in accordance with NDR-specified rules associated with specific NIEM concepts. Thus, an XML element is the child of an xs:appinfo and will not contain elements with the schema namespace.

###### NDR3 [Rule 9-81] (REF,EXT). Schema has data definition

[Rule 9-81](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-81), Schema has data definition (REF, EXT): [Section 9.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.7), Schema as a whole

**[OCL] context** InformationModel **inv:**

self.base\_Package.ownedComment.\_'body'->exists(doc|not(doc.oclIsUndefined())and(doc<>''))

###### NDR3 [Rule 9-85] (REF). No disallowed substitutions

[Rule 9-85](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-85), No disallowed substitutions (REF): [Section 9.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.7), Schema as a whole

[English]

The concept of disallowed substitutions (@blockDefault) is currently not supported by NIEM-UML. There will be no provisioning of the @blockDefault attribute.

###### NDR3 [Rule 9-86] (REF). No disallowed derivations

[Rule 9-86](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-86), No disallowed derivations (REF): [Section 9.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.7), Schema as a whole

[English]

The concept of disallowed derivations is currently not in the NIEM-UML model; the attribute @finalDefault will not be produced for any InformationModel schema.

###### NDR3 [Rule 9-87] (REF,EXT). No use of xs:redefine

[Rule 9-87](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-87), No use of xs:redefine (REF, EXT): [Section 9.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8), Schema assembly

[English]

The concept of xs:redefine is not in the NIEM-UML model; the schema construct xs:redefine can not be modeled and will not be produced for any InformationModel schema.

###### NDR3 [Rule 9-87] (REF,EXT). No use of xs:redefine2

[Rule 9-87](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-87), No use of xs:redefine (REF, EXT): [Section 9.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8), Schema assembly

[English]

The concept of xs:redefine is not in the NIEM-UML model; the schema construct xs:redefine can not be modeled and will not be produced for any InformationModel schema.

###### NDR3 [Rule 9-88] (REF,EXT). No use of xs:include

[Rule 9-88](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-88), No use of xs:include (REF, EXT): [Section 9.8](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.8), Schema assembly

[English]

The concept of xs:include is not in the NIEM-UML model; the schema construct xs:include can not be modeled and will not be produced for any InformationModel schema.

### <Stereotype> ReferenceName

##### Description

The ReferenceName stereotype is used on an Element that has a name that does not conform to the naming conventions required by the NIEM NDR or is otherwise not the desired NIEM name. The NIEMName attribute must provide a name for the Element that conforms to the relevant NDR naming rules for the specific kind of Element to which the stereotype is applied.

##### Extends

UML::Element

##### Properties

###### NIEMName : String [1]

A NIEM NDR-conformant name to be applied to an Element. The NIEMName will override any name generated from the UML name.

### <Stereotype> RoleOf

##### Description

The RoleOf stereotype is applied to a Property of a Class representing a NIEM role type, whose type identifies the base type of that role type. A RoleOf Property must be a reference (i.e., have aggregation=none). A NIEM role type is a type that represents a particular function, purpose, usage, or role of an object.

##### Extends

UML::Property

##### Constraints

###### NDR [Rule 7-40]

**[Rule 7-40] (REF, SUB, EXT)** Within the schema, any element with a name beginning with the string RoleOf SHALL represent a base type, of which the containing type represents a role.

**Rationale** A RoleOf element references its corresponding base element. The RoleOf label on the reference element ensures that a role object is distinguishable from other objects and its link to the associated base is also distinguishable from the additional properties that are characteristic of this role or that add information.

[English]

This constraint is implemented by the PIM/PSM transformation. Identifying a <Property> as a «RoleOf» corresponds to the NIEM naming convention used to identify the roleOf...reference and furthermore establishes the owning <Classifier> as a NIEM Role.

###### NDR [Rule 9-35]

**[Rule 9-35] (REF, SUB, EXT)** Within the schema, the name of a role SHALL use the property term RoleOf.

**Rationale** Using the property term RoleOf immediately identifies an element as representing a role.

[English]

This constraint is enforced by the PIM/PSM transformation. The transformation ensures that the "RoleOf" property term becomes part of the target PSM property name.

###### NDR3 [Rule 10-3] (REF,EXT). RoleOf element type is an object type

[Rule 10-3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-3), RoleOf element type is an object type (REF, EXT): [Section 10.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.2), Role types and roles

**[OCL] context** RoleOf **inv:**

(  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 and   
 not(self.type.oclIsUndefined())  
 and   
 not(self.name.oclIsUndefined())  
 and  
 (self.name.startsWith('RoleOf') or self.stereotypedBy('RolePlayedBy'))  
)   
implies  
self.type->forAll(t|not(t.stereotypedBy('MetadataType')or t.stereotypedBy('AssociationType')or t.stereotypedBy('AugmentationType') or t.oclIsKindOf(AssociationClass)))

### <Stereotype> RolePlayedBy

##### Description

RolePlayedBy Generalization specifies that the special class is to be considered the type of a role that is played by instances of the general class. In the PSM this will map to a property with the "RoleOf" prefix.

##### Extends

UML::Generalization

##### Constraints

###### NDR3 [Rule 10-3] (REF,EXT). RoleOf element type is an object type

[Rule 10-3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-3), RoleOf element type is an object type (REF, EXT): [Section 10.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.2), Role types and roles

**[OCL] context** RolePlayedBy **inv:**

self.base\_Generalization.general->  
 forAll(t|not(t.stereotypedBy('MetadataType')or t.stereotypedBy('AssociationType')or t.stereotypedBy('AugmentationType') or t.oclIsKindOf(AssociationClass)))

### <Stereotype> Subsets

##### Description

A Realization signifying a NIEM subsetting relationship between a client derived (subset) element and a supplier base (reference) element. The «Subsets» Realization must be between the same meta-types: either Properties, Classifiers, or «InformationModel» packages. The «Subsets» Realization must be between elements owned by different «InformationModel» packages. The targetNamespace of the distinct «InformationModel» packages must be identical. The defaultPurpose of client and supplier may be one of the following combinations: client is subset, supplier is reference; client is reference, supplier is reference; client is extension, supplier is extension; client is constraint, supplier is exchange, subset, extension, or reference

##### Generalization

[References](#_c7b8a68ef50d3d361f495647dd4876ec)

##### Constraints

###### SubsetNamesMustMatch

[English]

// pseudo code for specifying constraint in terms of tag values of foreign stereotype instances  
self.base\_Realization->forAll(r|  
 (r.client->forAll(c|  
 r.supplier->forAll(s|  
 ( s.name=c.name)  
 and (s.oclIsKindOf(Package) implies (c.oclIsKindOf(Package) and (s.getTargetNamespace()=c.getTargetNamespace())))  
 and (s.oclIsKindOf(Classifier) implies (c.oclIsKindOf(Classifier) and (s.getNearestPackage().getTargetNamespace()=c.getNearestPackage().getTargetNamespace())))  
 and (s.oclIsKindOf(Property) implies (c.oclIsKindOf(Property) and (s.getNearestPackage().getTargetNamespace()=c.getNearestPackage().getTargetNamespace())))  
   
 )  
 )  
 )  
)

### <Enumeration> DefaultPurposeCode

##### Description

The possible purposes for an information model. This enumeration provides the allowed values for the defaultPurpose attribute of the InformationModel stereotype. The values correspond to the schema purpose codes for an MPD artifact.

##### Literals

###### subset

A NIEM *schema document subset* is a set of XML schema documents that constitutes a reduced set of components derived from a NIEM reference schema document or document set associated with a given numbered release or domain update.

[Definition: *schema document subset*]

An XML schema document set based on a reference schema document set intended to ensure that any instance XML document valid to the schema document subset is also valid to the reference schema document set.

The primary purpose for a schema document subset is to reduce and constrain the scope and size of a full NIEM reference schema document set for use within an IEPD. A schema document subset is derived from a reference schema document set (such as a NIEM release) by applying subset operations. Also, note that employing a subset of a reference schema document set within an IEPD is optional; it is completely valid to reuse NIEM reference schema documents as-is within IEPDs.

The fundamental rule for a valid NIEM schema document subset is formally stated follows:

Rule 4-1. Fundamental NIEM Subset Rule

[Rule 4-1] (Schema-subset) (Constraint)

A schema document subset ($SUBSET) for a given reference schema document set ($REFERENCE) MUST be defined such that for all instance XML documents ($XML), where $XML is valid to $SUBSET, $XML is valid to $REFERENCE.

A schema document subset is composed of XML schema documents. A schema document subset can essentially be a reference schema document set (i.e., a NIEM release) that has been modified by applying subset operations to support business requirements represented in an IEPD. A subset derived from a reference schema document set may differ from that reference such that its content has been reduced and/or constrained.

[Definition: *subset schema document*]

An XML schema document that meets all of the following criteria:

* It is built from a reference schema document set where one or more reference schema documents have been substituted by corresponding subset schema documents.
* It is built from a reference schema document by applying subset operations to the XML schema statements in a reference schema document.
* It is explicitly designated as a subset schema document. This is accomplished by declaration in the relevant MPD catalog or by a tool-specific mechanism outside the subset schema document.
* It has a target namespace previously defined by a reference schema document. That is, it does not provide original definitions and declarations for schema components, but instead provides an alternate schema representation of components that are defined by a reference schema document.
* It does not alter the business semantics of components in its namespace. The reference schema document defines these business semantics.
* It is intended to express the limited vocabulary necessary for an IEPD and to support XML Schema validation for an IEPD.

 NIEM subset operations are essentially reduction operations that remove or constrain portions of a reference schema document set, thereby building a profile of the set. They do not expand the scope (i.e., relax constraints) or change the semantics of reference schema document set content.

Because NIEM adopts an optional and over-inclusive data representation strategy, most elements in a NIEM reference schema have zero to unbounded cardinality. So, elements with cardinality minOccurs="0" are optional and may be omitted from a subset schema document if not needed for business reasons. It is also valid to constrain element cardinality within a subset schema document, as long as doing so does not break the subset relationship with the reference schema document set. For example, a reference schema document element with cardinality (minOccurs="0", maxOccurs="unbounded") may be constrained to (0,1) or (1,1) in a subset schema document. However, if a reference schema document element’s cardinality is (1,unbounded), it may not be constrained to (0,1) since this breaks the subset relationship. The interval (0,1) is not contained within, and instead, overlaps the interval (1,unbounded).

The following list describes valid subset operations that are considered non-normative and informative only. In most cases, they can be applied to a schema document set and result in a corresponding schema document subset. However, it is possible to apply them in combinations that will break the subset relationship, or even result in invalid schemas. Apply these operations carefully and thoughtfully!

* 1. Remove an XML comment.
  2. Remove an xs:annotation and its children xs:documentation and xs:appinfo.
  3. Increase the value of an xs:element/@minOccurs as long as it remains less than or equal to its corresponding @maxOccurs value).
  4. Decrease the value of an xs:element/@maxOccurs as long as it remains greater than or equal to its corresponding @minOccurs value.
  5. Remove an xs:element if its @minOccurs="0".
  6. Remove an xs:complexType or xs:simpleType if not supporting an xs:element or xs:attribute declaration, or another xs:complexType or xs:simpleType definition.
  7. Remove an xs:attribute with @use="optional" from an xs:complexType.
  8. Change an xs:attribute/@use="optional" to @use="prohibited".
  9. Change an xs:attribute/@use="optional" to @use="required".
  10. Remove an xs:element declaration if it is not supporting an element use.
  11. Remove an xs:enumeration from an xs:simpleType as long as it is not the only remaining xs:enumeration.
  12. Remove an element with representation term AugmentationPoint if it is not being used for element substitution.
  13. Add or apply a constraining facet to an xs:simpleType.
  14. Remove an xs:import and its associated schema document if the schema document is not used within the document set.
  15. Change a concrete xs:element declaration to @abstract="true".
  16. Change an xs:element/@nillable="true" to @nillable="false".
  17. Substitute an xs:element/@substitutionGroup member for its associated substitution group head.
  18. Substitute a composition of xs:element/@substitutionGroup members for their associated substitution head (subject to cardinality and unique particle attribution (UPA) constraints[http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/ - cos-nonambig](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/#cos-nonambig)). The composition is an ordered sequence of the @substitutionGroup member elements. Each substitute element may bound its cardinality such that the total cardinality sum is within the bounds of the @substitutionGroup head cardinality. Order and cardinality of the replacement sequence must conform to XML Schema UPA constraints.
  19. Replace a wildcard (subject to cardinality, UPA, and namespace constraints) with a composition, i.e., an ordered sequence of elements. Each element may further bound cardinality within the bounds of the wildcard. Order and cardinality of replacement sequence must conform to XML Schema UPA constraints. The namespace of each element must conform with namespace constraints specified by the wildcard (if any).

###### constraint

[Definition: *constraint schema document set*]

A set of related constraint schema documents that work together, such as a constraint schema document set built by adding constraints to a schema document subset.

A constraint schema document set is an XML schema document set that is used to express business rules for a class of instance XML document, and is not assumed to be a definition for the semantics of the components it contains and describes. Instead, a constraint schema document set uses the XML Schema Definition Language to add constraints to components defined or declared by other schema documents, usually a schema document subset; but they can be applied to extension schema documents as well.

A constraint schema document set validates additional constraints imposed on an instance XML document only after it is known to be NIEM-conformant (i.e., has been validated with a reference schema document set, or schema document subset, and applicable extension schema documents).

To use a constraint schema document set to tighten constraints on an IEP, a two-pass validation technique is employed. In the first pass, an IEP is validated against the schema document subset and extension schema documents. This pass ensures that IEP semantics and structure conform to the NIEM model and NDR. In the second pass, an IEP is checked against a constraint schema document set, which may contain constrained versions of the subset schema documents and extension schema documents. This pass ensures that the IEP also satisfies the additional constraints (i.e., business rules that the first pass was unable to validate). A constraint schema document set need not validate constraints that are applied by other schema documents.

Constraint schema document sets are generally useful when it is necessary to impose restrictions that are more complex than cardinality. If only cardinality restrictions are needed, then it is easier and more efficient to set these directly in the subset schema documents and avoid the use of a constraint schema document set. Otherwise, a constraint schema document set may be necessary.

Use of a constraint schema document set is one option for tightening constraints on NIEM IEPs beyond what NIEM itself provides. This particular technique uses the XML Schema Definition Language. NIEM also allows other methods that do not use XML Schema. For example, the use of ISO Schematron is the preferred method for applying business rules. However, other constraint or business rule methods are also acceptable. That said, at this time there are no normative rules for how these business rule techniques should be employed in NIEM IEPDs. Therefore, if other techniques are used, it is a developer responsibility to incorporate appropriate artifacts and clear documentation.

Note that one disadvantage to use of constraint schema document sets is that they do not provide clear visibility or explanation of the constraints they enforce; nor do they provide clear validation failure messages. On the other hand, a standard business rule language such as ISO Schematron provides facilities for better understanding of the business rules, their intent, and error handling of failures.

A common practice for creating an IEPD constraint schema document set is to start with a valid NIEM schema document subset and modify it to further restrict the class of instance XML documents (IEPs) that will validate with this constraint schema set. However, an extension schema document can also be used to derive a constraint schema document. Using this technique, the namespace of that schema document would reuse the target namespace of the schema document from which it is derived.

There is no restriction on the number of constraint schema document sets (and passes) that an IEPD can employ. As in other advanced situations, developers must clearly document their intentions for and use of multiple constraint schema document sets.

In general, constraint schema documents in a constraint schema document set have far fewer requirements than other classes of NIEM schema documents. Since they work in tandem with NIEM normative schema documents, these schema documents are allowed to use the XML Schema Definition language in any way necessary to express business rules. This means that to constrain instance XML documents, these schema document can employ XML Schema constructs that are not allowed in NIEM conformant schema documents.

###### extension

[Definition: *extension schema document*]

An **extension schema document** is a schema docuemt that is intended to provide definitions of schema components that are intended for reuse within a more narrow scope than those defined by a reference schema document. It is a conformance target of this specification. An extension schema document MUST conform to all rules of this specification that apply to this conformance target. An XML document with a conformance target identifier of http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/#ExtensionSchemaDocument MUST be an extension schema document.

Characteristics of an extension schema document include:

* It is explicitly designated as an extension schema document via the conformance targets attribute.
* It provides the broadest, most fundamental definitions of components in its namespace.
* It provides the authoritative definition of business semantics for components in its namespace.
* It contains components that, when appropriate, use or are derived from the components in reference schema documents.
* It is intended to express the additional vocabulary required for an information exchange, above and beyond the vocabulary available from reference schemas, and to also support additional XML Schema validation requirements for an exchange.
* It satisfies all rules specified in this document for extension schema documents.

An extension schema in an information exchange specification serves several functions. First, it defines new content within a new namespace, which may be an exchange-specific namespace or a namespace shared by several exchanges. This content is NIEM-conformant but has fewer restrictions on it than do reference schema documents. Second, the extension schema document bases its content on content from reference schema documents, where appropriate. Methods of deriving content include using (by reference) existing schema components, as well as creating extensions and restrictions of existing components.

For example, an information exchange specification may define a type for an exchange-specific phone number and base that type on a type defined by the NIEM Core reference schema document. This exchange-specific phone number type may restrict the NIEM Core type to limit those possibilities that are permitted of the base type. Exchange extensions and restrictions must include annotations and documentation to be conformant, but they are allowed to use restriction, choice, and some other constructs that are not allowed in reference schema documents.

Note that exchange specifications may define schemas that meet the criteria of reference schemas for those components that its developers wish to nominate for later inclusion in NIEM Core or in domains.

From the MPD:

In general, an extension schema document contains components that use or are derived from the components in reference schema documents. It is intended to express additional vocabulary above and beyond the vocabulary available from reference schema documents.

A developer who determines that NIEM is missing elements required for a given information exchange has several options to account for such requirement shortfalls. Using rules and techniques defined in the NDR:

* + Extend an existing NIEM data component (if possible).
  + Augment an existing NIEM data type (through NIEM Type Augmentation).
  + Build a new NIEM-conformant data component.
  + Employ NIEM adapter types for components from an external standard that does not conform to NIEM.

A NIEM extension schema document may contain data components built from any of the options above. Employment of extension schema documents in an IEPD is entirely optional.

Multiple extension schema documents are allowed in a single IEPD. Developers will likely want to reuse many of their extension schema documents in other IEPDs. Therefore, the best practice for extension is to group all data components designed to be reused into one extension schema document or document set, and group IEPD-specific data components into another. Then the reusable extension components can be more easily redeployed in other IEPDs as needed.

Extension schema documents generally contain new data component declarations that may (though not necessarily) be derived from or reference existing NIEM data component. This being the case, reference schema documents do not exist for new data components found within extension schema documents. Therefore, extension schema documents must satisfy the more rigorous documentation requirements of reference schema documents. Per the NDR, the definition or declaration of each new data component in an extension schema document must include an xs:annotation element that provides its semantics and NIEM-specific relationships.

###### incremental

###### reference

[Definition: *reference schema document*]

A **reference schema document** is a schema document that is intended to provide the authoritative definitions of broadly reusable schema components. It is a conformance target of this specification. A reference schema document MUST conform to all rules of this specification that apply to this conformance target. An XML document with a conformance target identifier of http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/#ReferenceSchemaDocument MUST be a conformant reference schema document.

A reference schema document is a schema document that is intended to be the authoritative definition schema for a namespace. Examples include NIEM Core and NIEM domains.

Some characteristics of a reference schema document:

* It is explicitly designated as a reference schema via the conformance targets attribute, per Rule 4-5, [*Schema claims reference schema conformance target*](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_4-5) *(REF).*
* It provides the broadest, most fundamental definitions of components in its namespace.
* It provides the authoritative definition of business semantics for components in its namespace.
* It is intended to serve as the basis for components in information exchanges and extension schema documents.
* It satisfies all rules specified in the Naming and Design Rules for reference schemas.

Any schema that defines components that are intended to be incorporated into NIEM Core or a NIEM domain will be defined as a reference schema.

The rules for reference schema documents are more stringent than are the rules for other classes of NIEM-conformant schemas. Reference schema documents are intended to support the broadest reuse. They are very uniform in their structure. As they are the primary definitions for schema components, they do not need to restrict other data definitions, and they are not allowed to use XML Schema’s restriction mechanism (e.g., Rule 9-29, [*Complex content uses extension*](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-29) *(REF)*). Reference schema documents are intended to be as regular and simple as possible.

From the MPD:

A reference schema document generally applies to NIEM releases, core updates, and domain updates. Though not common, it is also valid to use a reference schema document or document set within an IEPD.

A NIEM reference schema document is intended to be the authoritative definition schema document for a NIEM target namespace, therefore, all NIEM releases, core updates, and domain updates are composed of a reference schema document set and associated namespaces. As a standalone artifact set, a reference schema document set is always harmonized such that all types and properties are semantically unique (i.e., multiple versions of semantically identical types or properties do not exist within the set).

As authoritative definitions, NIEM reference schema document sets satisfy more rigorous documentation requirements.  Typically reference schema documents contain data components with the most relaxed cardinality (zero to unbounded). However, this is not an absolute requirement. If necessary, cardinality in reference schema documents may be constrained to model reality. For example, in NIEM 3.0 a nc:Location2DGeospatialCoordinateType contains both a nc:GeographicCoordinateLatitude element and a nc:GeographicCoordinateLongitude element. Each of these elements has cardinality minOccurs="1" and maxOccurs="1". Any other cardinality for these elements has no meaning. On the other hand, one might claim that NIEM should constrain nc:PersonType to a single occurrence of the element nc:PersonBirthDate. Every person has one and only one birth date. Unfortunately, also in reality, criminal persons often present multiple identities with multiple birth dates; and so the capability to represent such is an important data requirement for NIEM.

###### replacement

###### external

[Definition: *external schema document*]

Any XML schema document that is not one of:

1. a reference schema document,
2. an extension schema document, or
3. an XML schema document that has the structures namespace as its target namespace.

All MPD classes may contain *external schema documents* that do not conform to NIEM. Data components declared and defined in external schema documents require NIEM *external adapter types* to identify the fact they do not conform to NIEM.

[Definition: *external adapter type*]

A NIEM-conformant type that adapts external components for use within NIEM. An external adapter type creates a new class of object that embodies a single concept composed of external components. A NIEM-conformant schema defines an external adapter type.

## Profile : NIEM\_PSM\_Profile

### Overview

The NIEM PSM Profile comprises stereotypes that are used in NIEM PSMs. These stereotypes need not be used with a NIEM PIM, but they may be in order to provide additional platform-specific markup.Further, the NIEM PIM Profile imports the NIEM Common Profile and, therefore, includes all the stereotypes and metaclasses covered by that profile.

### <Stereotype> XSDAnyProperty

##### Description

XSDAnyProperty stereotype represents a property that is unrestricted with respect to its type, which is implemented in XML Schema as the xsd:any particle.

##### Extends

UML::Property

##### Properties

###### processContents : XSDProcessContentsCode [1]

Determines how or if the value of a NIEM property should be processed; values are: "lax", "skip", and "strict".

###### valueNamespace : String [1]

The namespace in which values of this property must be defined. Implemented in XML Schema as the value of the namespace attribute on the xsd:any element.

##### Constraints

###### XSDAnyPropertyType

An XSDAnyProperty must have an empty type and must not be a derived union or subset any other property.

**[OCL] context** XSDAnyProperty **inv:**

self.base\_Property.type.oclIsUndefined() and  
 not(self.base\_Property.isDerivedUnion) and  
 self.base\_Property.subsettedProperty->isEmpty()

### <Stereotype> XSDDeclaration

##### Description

The XSDDeclaration stereotype is a specialization of the common References stereotype. However, it is constrained such that its client must be an XSDProperty Property and its supplier must be an XSDProperty Property or a Namepsace Package. By default, the namespace of the global XSD property declaration referenced by XSDProperty is the namespace of its class. The XSDDeclaration stereotype allows the modeler to specify the namespace a XSDProperty will reference based on the namespace of another XSDProperty or the target namespace of a Namespace Package. Specifically, the client of the XSDDeclaration Realization shall reference the namespace indicated by the supplier of the XSDDeclaration Realization, the client of the maps to one of the following: an attribute use schema component or a particle component whose term property is an element declaration schema component. In the first case, the supplier maps to the attribute declaration schema component for the attribute use component. In the second case, the supplier maps to the element declaration schema component for the particle schema component.

##### Generalization

[References](#_c7b8a68ef50d3d361f495647dd4876ec)

### <Stereotype> XSDProperty

##### Description

An XSDProperty Property represents a NIEM property, which is implemented in XML Schema as either an attribute declaration and use or an element declaration and particle. If an XSDProperty Property is the client of a References Realization, then the supplier of the Realization defines the declaration of the NIEM property. Otherwise, the declaration of the NIEM property is defined implicitly to be the top-level attribute or element definition of the same name within the target namespace of the Namespace Package that contains the XSDProperty Property. All NIEM properties represented by XSDProperty Properties with the same name within the same package that are not clients of References Realizations share the same implicit attribute or element declaration.

##### Extends

UML::Property

##### Properties

###### fixed : String [0..1]

If present, implemented as the value of the fixed attribute of the xsd:attribute or xsd:element.

###### kind : XSDPropertyKindCode [1]

Indicates whether the NIEM property is implemented in XML Schema as an attribute declaration and attribute use or element declaration and element particle: if "attribute", the NIEM property is implemented in XML Schema as an attribute declaration and attribute use; if "element", the NIEM property is implemented as an element declaration and element particle.

###### nillable : Boolean [0..1]

Implemented in XML Schema as the value of the nillable attribute on the xsd:element element. Note that an XSDProperty that represents an XML attribute may not have a nillable value.

##### Constraints

###### NDR3 [Rule 10-10] (REF,EXT). Element use from external adapter type defined by external schema documents

[Rule 10-10](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-10), Element use from external adapter type defined by external schema documents (REF, EXT): [Section 10.2.3.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.2), External adapter types

**[OCL] context** XSDProperty **inv:**

(  
 self.namespace.stereotypedBy('AdapterType')  
 and  
 (not(self.stereotypedBy('XSDProperty')) or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element))  
)  
   
implies  
self.clientDependency->select(r|r.stereotypedBy('References')).supplier  
->forAll(s|s.namespace.namespace.appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel).defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::external)

###### NDR3 [Rule 10-13] (REF). External attribute use only in external adapter type

[Rule 10-13](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-13), External attribute use only in external adapter type (REF): [Section 10.2.3.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.3), External attribute use

**[OCL] context** XSDProperty **inv:**

(  
 (self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)   
 and  
 self.base\_Property.namespace.namespace.appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel)  
 ->forAll(im|(im.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::subset)or(im.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference))  
   
 and  
 not(self.base\_Property.namespace.stereotypedBy('AdapterType'))   
)   
implies  
self.base\_Property.clientDependency->select(d|d.stereotypedBy('References')).supplier.namespace.namespace.appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel)  
 ->forAll(typeInformationModel|typeInformationModel.defaultPurpose<>NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::external)

###### NDR3 [Rule 10-14] (REF,EXT). External attribute use has data definition

[Rule 10-14](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-14), External attribute use has data definition (REF, EXT): [Section 10.2.3.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.3), External attribute use

**[OCL] context** XSDProperty **inv:**

(  
 (self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)  
 and  
 self.base\_Property.clientDependency->select(d|d.stereotypedBy('References')).supplier.namespace.namespace  
 ->forAll(typeInformationModel|typeInformationModel.appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel).defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::external)  
)   
implies  
self.base\_Property.ownedComment.\_'body'->exists(b|not(b.oclIsUndefined()) and b<>'')

###### NDR3 [Rule 10-15] (SET). External attribute use not an ID

[Rule 10-15](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-15), External attribute use not an ID (SET): [Section 10.2.3.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.3), External attribute use

**[OCL] context** XSDProperty **inv:**

(  
 (self.kind.oclAsType(EnumerationLiteral).name='attribute')  
 and not(self.base\_Property.type.oclIsUndefined())  
 and not(self.base\_Property.type.name.oclIsUndefined())  
 and not(self.base\_Property.type.namespace.oclIsUndefined())  
 and not(self.base\_Property.type.namespace.oclAsType(NamedElement).name.oclIsUndefined())  
)  
implies   
(   
 (self.base\_Property.type.name<>'ID')   
 and (self.base\_Property.type.namespace.oclAsType(NamedElement).name<>'XMLPrimitiveTypes')  
)

###### NDR3 [Rule 10-16] (REF,EXT). External element use has data definition

[Rule 10-16](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-16), External element use has data definition (REF, EXT): [Section 10.2.3.4](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.2.3.4), External element use

**[OCL] context** XSDProperty **inv:**

(  
   
 (not(self.stereotypedBy('XSDProperty')) or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element))  
 and   
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 and  
 self.clientDependency->select(d|d.stereotypedBy('References'))->exists(d|d.supplier.namespace.namespace  
 ->exists(typeInformationModel|typeInformationModel.appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel).defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::external)  
 )   
)   
implies(  
 self.ownedComment.\_'body'->exists(b|not(b.oclIsUndefined()) and b<>'')  
)

###### NDR3 [Rule 10-20] (REF,EXT). Association element type is an association type

[Rule 10-20](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-20), Association element type is an association type (REF, EXT): [Section 10.3.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.3.2), Association element declarations

**[OCL] context** XSDProperty **inv:**

(   
 (  
 not(self.namespace.oclIsUndefined())  
 and not(self.namespace.namespace.oclIsUndefined())  
 and self.namespace.namespace.stereotypedBy('InformationModel')  
 and  
 (not(self.stereotypedBy('XSDProperty')) or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind.oclAsType(EnumerationLiteral).name='element'))  
 and not(self.name.oclIsUndefined())  
 and self.name.endsWith('Association')  
 and not(self.type.oclIsUndefined())  
 )   
 implies  
 (self.type.stereotypedBy('AssociationType')or self.type.oclIsKindOf(AssociationClass))  
)  
and  
(  
 (  
 not(self.namespace.oclIsUndefined())  
 and not(self.namespace.namespace.oclIsUndefined())  
 and self.namespace.namespace.stereotypedBy('InformationModel')  
 and  
 (not(self.stereotypedBy('XSDProperty')) or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind.oclAsType(EnumerationLiteral).name='element'))  
 and not(self.type.oclIsUndefined())  
 and (self.type.stereotypedBy('AssociationType')or self.type.oclIsKindOf(AssociationClass))  
 and not(self.name.oclIsUndefined())   
 and (self.name<>'')  
)   
 implies  
 self.name.endsWith('Association')  
)

###### NDR3 [Rule 10-34] (REF,EXT). Augmentation element type is an augmentation type

[Rule 10-34](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-34), Augmentation element type is an augmentation type (REF, EXT): [Section 10.4.5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.5), Augmentation element declarations

**[OCL] context** XSDProperty **inv:**

(  
 (  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 and  
 (not(self.stereotypedBy('XSDProperty')) or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element))  
 and  
 not(self.name.oclIsUndefined())  
 and  
 self.name.endsWith('Augmentation')  
 and  
 not(self.type.oclIsUndefined())  
 )   
 implies  
 self.type.stereotypedBy('AugmentationType')  
)  
and  
(  
 (  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 and  
 (not(self.stereotypedBy('XSDProperty')) or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element))  
 and  
 not(self.type.oclIsUndefined())  
 and  
 self.type.stereotypedBy('AugmentationType')  
 )   
 implies  
 self.name.endsWith('Augmentation')  
)

###### NDR3 [Rule 10-35] (REF,SET). Augmentation elements are not used directly

[Rule 10-35](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-35), Augmentation elements are not used directly (REF, SET): [Section 10.4.5](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.4.5), Augmentation element declarations

**[OCL] context** XSDProperty **inv:**

(  
 (not(self.stereotypedBy('XSDProperty')) or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element))  
 and  
 (not(self.name.oclIsUndefined()) and self.name.endsWith('Augmentation'))  
 and   
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 )   
 implies  
   
 (  
 self.namespace->forAll(t|t.stereotypedBy('PropertyHolder'))  
 )

###### NDR3 [Rule 10-39] (REF,EXT). Metadata element declaration type is a metadata type

[Rule 10-39](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-39), Metadata element declaration type is a metadata type (REF, EXT): [Section 10.5.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.5.2), Metadata element declarations

**[OCL] context** XSDProperty **inv:**

(  
 (  
 (  
 not(self.stereotypedBy('XSDProperty'))   
 or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind.oclAsType(EnumerationLiteral).name='element')  
 )  
 and self.name.endsWith('Metadata')  
 and not(self.namespace.oclIsUndefined())  
 and not(self.namespace.namespace.oclIsUndefined())  
 and self.namespace.namespace.stereotypedBy('InformationModel')  
 )   
 implies  
 (not(self.type.oclIsUndefined()) and self.type.stereotypedBy('MetadataType'))  
)  
and  
(  
 (  
 (  
 not(self.stereotypedBy('XSDProperty'))   
 or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind.oclAsType(InstanceSpecification).name='element')  
 )  
 and not(self.type.oclIsUndefined())   
 and not(self.name.oclIsUndefined())   
 and self.type.stereotypedBy('MetadataType')  
 and not(self.namespace.oclIsUndefined())  
 and not(self.namespace.namespace.oclIsUndefined())  
 and self.namespace.namespace.stereotypedBy('InformationModel')  
 )   
 implies  
 self.name.endsWith('Metadata')  
)

###### NDR3 [Rule 10-40] (REF,EXT,SET). Metadata element has applicable elements

[Rule 10-40](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-40), Metadata element has applicable elements (REF, EXT, SET): [Section 10.5.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.5.2), Metadata element declarations

[English]

The rule is definitional.

###### NDR3 [Rule 10-47] (REF,EXT). Attribute name begins with lower case letter

[Rule 10-47](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-47), Attribute name begins with lower case letter (REF, EXT): [Section 10.8.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.1), Character case

**[OCL] context** XSDProperty **inv:**

(self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute) implies  
 self.base\_Property.name.firstToUpper()<>self.base\_Property.name

###### NDR3 [Rule 10-62] (REF,EXT). Element with simple content has representation term

[Rule 10-62](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-62), Element with simple content has representation term (REF, EXT): [Section 10.8.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.7), Representation terms

**[OCL] context** XSDProperty **inv:**

if(  
 not(self.namespace.oclIsUndefined())  
 and not(self.namespace.stereotypedBy('List'))  
 and not(self.namespace.namespace.oclIsUndefined())  
 and self.namespace.namespace.stereotypedBy('InformationModel')  
 and not( self.type.oclIsUndefined())   
 and not(self.name.oclIsUndefined())  
 and self.type.oclIsKindOf(DataType)   
 and (  
 not(self.stereotypedBy('XSDProperty'))   
 or   
 (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind.oclAsType(EnumerationLiteral).name='element')   
 )  
 )  
then (  
 self.name.match('.\*Name.\*') or  
 self.name.match('.\*Text.\*') or  
 self.name.match('.\*List.\*') or  
 self.name.match('.\*Quantity.\*') or  
 self.name.match('.\*Percent.\*') or  
 self.name.match('.\*Rate.\*') or  
 self.name.match('.\*Value.\*') or  
 self.name.match('.\*Numeric.\*') or  
 self.name.match('.\*Measure.\*') or  
 self.name.match('.\*Indicator.\*') or  
 self.name.match('.\*URI.\*')   
 or self.name.match('.\*ID.\*')  
 or self.name.match('.\*Time.\*') or  
 self.name.match('.\*Date.\*') or  
 self.name.match('.\*Duration.\*') or  
 self.name.match('.\*DateTime.\*') or  
 self.name.match('.\*Code.\*') or  
 self.name.match('.\*Video.\*') or  
 self.name.match('.\*Sound.\*') or  
 self.name.match('.\*Picture.\*') or  
 self.name.match('.\*Graphic.\*') or  
 self.name.match('.\*BinaryObject.\*') or  
 self.name.match('.\*Amount.\*')   
 )   
 else(true)endif

###### NDR3 [Rule 10-63] (REF,EXT). Name has representation term when appropriate

[Rule 10-63](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-63), Name has representation term when appropriate (REF, EXT): [Section 10.8.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.7), Representation terms

[English]

Constraint is non-computable.

###### NDR3 [Rule 10-64] (REF,EXT). Name has representation term only when appropriate

[Rule 10-64](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_10-64), Name has representation term only when appropriate (REF, EXT): [Section 10.8.7](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_10.8.7), Representation terms

[English]

Constraint is non-computable.

###### NDR3 [Rule 11-12] (REF,EXT). Element name is upper camel case

[Rule 11-12](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-12), Element name is upper camel case (REF, EXT): [Section 11.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.2.1), Element declaration

**[OCL] context** XSDProperty **inv:**

(   
 not(self.name.oclIsUndefined())  
 and(self.name<>'')  
 and not(self.namespace.oclIsUndefined())  
 and not(self.namespace.stereotypedBy('List'))  
 and not(self.namespace.namespace.oclIsUndefined())  
 and self.namespace.namespace.stereotypedBy('InformationModel')  
 and  
 (  
 not(self.stereotypedBy('XSDProperty'))  
 or  
 (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind.oclAsType(EnumerationLiteral)='element')  
 )   
 )   
implies  
self.name.match('^([A-Z][A-Za-z0-9\\-]\*)+$')

###### NDR3 [Rule 11-13] (REF,EXT). Element type does not have a simple type name

[Rule 11-13](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-13), Element type does not have a simple type name (REF, EXT): [Section 11.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.2.1), Element declaration

**[OCL] context** XSDProperty **inv:**

(  
 (  
 not(self.stereotypedBy('XSDProperty'))  
 or  
 (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind.oclAsType(EnumerationLiteral)='element')  
 )   
 and not(self.type.oclIsUndefined())  
 and not(self.type.name.oclIsUndefined())  
 and  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 )   
implies  
not(self.type.name.endsWith('SimpleType'))

###### NDR3 [Rule 11-14] (REF,EXT). Element type is from conformant namespace

[Rule 11-14](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-14), Element type is from conformant namespace (REF, EXT): [Section 11.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.2.1), Element declaration

**[OCL] context** XSDProperty **inv:**

(  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 and  
 not(self.type.oclIsUndefined())   
 and self.type.namespace.stereotypedBy('Namespace')  
 and  
 (  
 not(self.stereotypedBy('XSDProperty'))  
 or  
 (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind.oclAsType(EnumerationLiteral).name='element')  
 )   
 )   
implies   
self.type.namespace.appliedStereotype('Namespace').oclAsType(NIEM\_UML\_Profile::NIEM\_Common\_Profile::Namespace).isConformant

###### NDR3 [Rule 11-15] (REF,EXT). Name of element that ends in "Abstract" is abstract

[Rule 11-15](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-15), Name of element that ends in Abstract is abstract (REF, EXT): [Section 11.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.2.1), Element declaration

**[OCL] context** XSDProperty **inv:**

(  
 (  
 not(self.stereotypedBy('XSDProperty'))  
 or  
 (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element)  
 )   
 and  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')   
 and  
 self.name.endsWith('Abstract')  
 )   
implies   
self.isDerivedUnion

###### NDR3 [Rule 11-16] (REF,EXT). Name of element declaration with simple content has representation term

[Rule 11-16](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-16), Name of element declaration with simple content has representation term (REF, EXT): [Section 11.2.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.2.1.1), Object element declarations

**[OCL] context** XSDProperty **inv:**

(  
 (  
 not(self.stereotypedBy('XSDProperty'))  
 or  
 (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind.oclAsType(EnumerationLiteral).name='element')  
 )   
 and not(self.type.oclIsUndefined())  
 and self.type.oclIsKindOf(DataType)   
 and not(self.namespace.oclIsUndefined())  
 and not(self.namespace.namespace.oclIsUndefined())  
 and self.namespace.namespace.stereotypedBy('InformationModel')  
 )   
implies (  
 self.name.match('.\*Name.\*') or  
 self.name.match('.\*Text.\*') or  
 self.name.match('.\*List.\*') or  
 self.name.match('.\*Quantity.\*') or  
 self.name.match('.\*Percent.\*') or  
 self.name.match('.\*Rate.\*') or  
 self.name.match('.\*Value.\*') or  
 self.name.match('.\*Numeric.\*') or  
 self.name.match('.\*Measure.\*') or  
 self.name.match('.\*Indicator.\*') or  
 self.name.match('.\*URI.\*')   
 or self.name.match('.\*ID.\*')  
 or self.name.match('.\*Time.\*') or  
 self.name.match('.\*Date.\*') or  
 self.name.match('.\*Duration.\*') or  
 self.name.match('.\*DateTime.\*') or  
 self.name.match('.\*Code.\*') or  
 self.name.match('.\*Video.\*') or  
 self.name.match('.\*Sound.\*') or  
 self.name.match('.\*Picture.\*') or  
 self.name.match('.\*Graphic.\*') or  
 self.name.match('.\*BinaryObject.\*') or  
 self.name.match('.\*Amount.\*')   
 )

###### NDR3 [Rule 11-17] (SET). Name of element declaration with simple content has representation term

[Rule 11-17](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-17), Name of element declaration with simple content has representation term (SET): [Section 11.2.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.2.1.1), Object element declarations

**[OCL] context** XSDProperty **inv:**

(  
 (  
 not(self.stereotypedBy('XSDProperty'))  
 or  
 (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element)  
 )   
 and   
 (not(self.type.oclIsUndefined()) and self.type.oclIsKindOf(DataType))   
 and  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 )   
implies (  
 self.name.match('.\*Name.\*') or  
 self.name.match('.\*Text.\*') or  
 self.name.match('.\*List.\*') or  
 self.name.match('.\*Quantity.\*') or  
 self.name.match('.\*Percent.\*') or  
 self.name.match('.\*Rate.\*') or  
 self.name.match('.\*Value.\*') or  
 self.name.match('.\*Numeric.\*') or  
 self.name.match('.\*Measure.\*') or  
 self.name.match('.\*Indicator.\*') or  
 self.name.match('.\*URI.\*')   
 or self.name.match('.\*ID.\*')  
 or self.name.match('.\*Time.\*') or  
 self.name.match('.\*Date.\*') or  
 self.name.match('.\*Duration.\*') or  
 self.name.match('.\*DateTime.\*') or  
 self.name.match('.\*Code.\*') or  
 self.name.match('.\*Video.\*') or  
 self.name.match('.\*Sound.\*') or  
 self.name.match('.\*Picture.\*') or  
 self.name.match('.\*Graphic.\*') or  
 self.name.match('.\*BinaryObject.\*') or  
 self.name.match('.\*Amount.\*')   
 )

###### NDR3 [Rule 11-18] (REF,EXT). Element substitution group defined by conformant schema

[Rule 11-18](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-18), Element substitution group defined by conformant schema (REF, EXT): [Section 11.2.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.2.2), Element substitution group

**[OCL] context** XSDProperty **inv:**

(  
 (  
 not(self.stereotypedBy('XSDProperty'))  
 or  
 (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element)  
 )   
 and  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 )   
implies   
self.subsettedProperty.namespace.namespace->forAll(m|m.appliedStereotype('Namespace').oclAsType(NIEM\_UML\_Profile::NIEM\_Common\_Profile::Namespace).isConformant)

###### NDR3 [Rule 11-19] (REF,EXT). Attribute type defined by conformant schema

[Rule 11-19](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-19), Attribute type defined by conformant schema (REF, EXT): [Section 11.2.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.2.3), Attribute declaration

**[OCL] context** XSDProperty **inv:**

(  
 (self.kind.oclAsType(EnumerationLiteral)='attribute')   
 and(not(self.base\_Property.type.oclIsUndefined()))  
 and(not(self.base\_Property.type.namespace.oclIsUndefined()))  
 and(not(self.base\_Property.type.namespace.name.oclIsUndefined()))  
 and (  
 self.base\_Property.type.namespace.stereotypedBy('InformationModel')  
 or (self.base\_Property.type.namespace.name='XMLPrimitiveTypes')  
 )   
)   
implies(   
 (self.base\_Property.type.namespace.name='XMLPrimitiveTypes')  
 or  
 self.base\_Property.type.namespace.appliedStereotype('Namespace').oclAsType(NIEM\_UML\_Profile::NIEM\_Common\_Profile::Namespace).isConformant  
)

###### NDR3 [Rule 11-20] (REF,EXT). Attribute name uses representation term

[Rule 11-20](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-20), Attribute name uses representation term (REF, EXT): [Section 11.2.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.2.3), Attribute declaration

**[OCL] context** XSDProperty **inv:**

(self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)   
implies(  
 self.base\_Property.name.endsWith('List') or  
 self.base\_Property.name.endsWith('Name') or  
 self.base\_Property.name.endsWith('Text') or  
 self.base\_Property.name.endsWith('Quantity') or  
 self.base\_Property.name.endsWith('Percent') or  
 self.base\_Property.name.endsWith('Rate') or  
 self.base\_Property.name.endsWith('Value') or  
 self.base\_Property.name.endsWith('Numeric') or  
 self.base\_Property.name.endsWith('Measure') or  
 self.base\_Property.name.endsWith('Indicator') or  
 self.base\_Property.name.endsWith('URI') or   
 self.base\_Property.name.endsWith('ID')  
 or self.base\_Property.name.endsWith('Time') or  
 self.base\_Property.name.endsWith('Duration') or  
 self.base\_Property.name.endsWith('Date') or  
 self.base\_Property.name.endsWith('DateTime') or  
 self.base\_Property.name.endsWith('Code') or  
 self.base\_Property.name.endsWith('Video') or  
 self.base\_Property.name.endsWith('Sound') or  
 self.base\_Property.name.endsWith('Picture') or  
 self.base\_Property.name.endsWith('Graphic') or  
 self.base\_Property.name.endsWith('BinaryObject') or  
 self.base\_Property.name.endsWith('Amount')   
)

###### NDR3 [Rule 11-21] (REF,EXT). Element or attribute declaration introduced only once into a type

[Rule 11-21](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-21), Element or attribute declaration introduced only once into a type (REF, EXT): [Section 11.3.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.3.2.1), Element use

[English]

Satisfied by UML Constraint members\_distinguishable, where the Namespace is a Classifier and the members are Properties.

###### NDR3 [Rule 11-22] (REF,EXT). Element reference defined by conformant schema

[Rule 11-22](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-22), Element reference defined by conformant schema (REF, EXT): [Section 11.3.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.3.2.1), Element use

**[OCL] context** XSDProperty **inv:**

(  
 (  
 not(self.stereotypedBy('XSDProperty'))  
 or  
 (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element)  
 )   
 and  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 )   
implies   
self.clientDependency->select(d|d.stereotypedBy('References')).supplier->select(s|s.oclIsKindOf(Property)).namespace.namespace  
->forAll(m|m.appliedStereotype('Namespace').oclAsType(NIEM\_UML\_Profile::NIEM\_Common\_Profile::Namespace).isConformant)

###### NDR3 [Rule 11-23] (REF,EXT). Referenced attribute defined by conformant schemas

[Rule 11-23](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-23), Referenced attribute defined by conformant schemas (REF, EXT): [Section 11.3.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.3.3), Attribute use

**[OCL] context** XSDProperty **inv:**

(self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)  
implies   
self.base\_Property.clientDependency->select(d|d.stereotypedBy('References')).supplier->select(s|s.oclIsKindOf(Property)).namespace.namespace  
->forAll(m|m.appliedStereotype('Namespace').oclAsType(NIEM\_UML\_Profile::NIEM\_Common\_Profile::Namespace).isConformant)

###### NDR3 [Rule 11-31] (REF,EXT). Standard opening phrase for element

[Rule 11-31](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_11-31), Standard opening phrase for element (REF, EXT): [Section 11.6.1.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_11.6.1.1), Data definition opening phrases

**[OCL] context** XSDProperty **inv:**

(  
 (  
 not(self.stereotypedBy('XSDProperty'))  
 or  
 (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind.oclAsType(EnumerationLiteral).name='element')  
 )   
 and not(self.namespace.oclIsUndefined())  
 and not(self.namespace.namespace.oclIsUndefined())  
 and self.namespace.namespace.stereotypedBy('InformationModel')   
 and not(self.name.oclIsUndefined())  
 and (self.name<>'')  
 )   
implies   
if (self.name.endsWith('AugmentationPoint')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.startsWith('an augmentation point '))   
 else if (self.name.endsWith('Augmentation')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.startsWith('supplements ')or b.startsWith('additional information about '))   
 else if (self.name.endsWith('Metadata')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('(metadata about|information that further qualifies).\*'))   
 else if (self.name.endsWith('Association')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^an?( .\*)? (relationship|association).\*'))   
 else if (self.type.oclIsUndefined()) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('a data concept.\*'))   
 else if (self.name.endsWith('Date')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^an?( .\*)? (date|month|year).\*'))   
 else if (self.name.endsWith('Quantity')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^an?( .\*)? (count|number).\*'))   
 else if (self.name.endsWith('Picture')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^an?( .\*)? (image|picture|photograph).\*'))   
 else if (self.name.endsWith('Indicator')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^true if .\*; false (otherwise|if).\*'))   
 else if (self.name.endsWith('Identification')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^an?( .\*)? identification.\*'))   
 else if (self.name.endsWith('Name')) then self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('^(a|an)( .\*)? name.\*'))   
 else (self.ownedComment.\_'body'.toLower().normalizeSpace()->exists(b|b.match('(a|an).\*')))   
 endif  
 endif   
 endif  
 endif  
 endif  
 endif   
 endif   
 endif   
 endif   
 endif   
endif

###### NDR3 [Rule 9-37] (REF,EXT). Untyped element is abstract

[Rule 9-37](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-37), Untyped element is abstract (REF, EXT): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

**[OCL] context** XSDProperty **inv:**

(   
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')   
 and  
 (not(self.stereotypedBy('XSDProperty')) or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element))  
 )   
 implies  
(self.type.oclIsUndefined() implies self.isDerivedUnion)

###### NDR3 [Rule 9-38] (REF,EXT). Element of type xs:anySimpleType is abstract

[Rule 9-38](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-38), Element of type xs:anySimpleType is abstract (REF, EXT): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

**[OCL] context** XSDProperty **inv:**

(  
 (not(self.stereotypedBy('XSDProperty')) or (self.appliedStereotype('XSDProperty').oclAsType(NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDProperty).kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::element))  
 and  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
 and  
 not(self.type.oclIsUndefined())  
 and   
 (self.type.name='anySimpleType')  
 and   
 (self.type.\_'package'.name='XMLPrimitiveTypes')  
)   
implies   
self.isDerivedUnion

###### NDR3 [Rule 9-43] (REF). No element disallowed derivation

[Rule 9-43](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-43), No element disallowed derivation (REF): [Section 9.2.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.1), Element declaration

**[OCL] context** XSDProperty **inv:**

(  
 not(self.namespace.oclIsUndefined())  
 and  
 not(self.namespace.namespace.oclIsUndefined())  
 and  
 self.namespace.namespace.stereotypedBy('InformationModel')  
)  
implies  
(  
 self.namespace.namespace.appliedStereotype('InformationModel').oclAsType(NIEM\_UML\_Profile::NIEM\_PIM\_Profile::InformationModel)  
 ->forAll(im|  
 (im.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::reference)  
 or(im.defaultPurpose=NIEM\_UML\_Profile::NIEM\_PIM\_Profile::DefaultPurposeCode::subset)  
 )  
 implies  
 not(self.isLeaf)  
)

###### NDR3 [Rule 9-50] (REF,EXT). No attribute type of xs:ID

[Rule 9-50](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-50), No attribute type of xs:ID (REF, EXT): [Section 9.2.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3.1), Prohibited attribute types

**[OCL] context** XSDProperty **inv:**

(self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)   
implies   
self.base\_Property.type->forAll(t|not((t.name='ID')and(t.\_'package'.name='XMLPrimitiveTypes')))

###### NDR3 [Rule 9-51] (REF,EXT). No attribute type of xs:IDREF

[Rule 9-51](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-51), No attribute type of xs:IDREF (REF, EXT): [Section 9.2.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3.1), Prohibited attribute types

**[OCL] context** XSDProperty **inv:**

(self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)   
implies   
self.base\_Property.type->forAll(t|not((t.name='IDREF')and(t.\_'package'.name='XMLPrimitiveTypes')))

###### NDR3 [Rule 9-52] (REF,EXT). No attribute type of xs:IDREFS

[Rule 9-52](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-52), No attribute type of xs:IDREFS (REF, EXT): [Section 9.2.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3.1), Prohibited attribute types

**[OCL] context** XSDProperty **inv:**

(self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)   
implies   
self.base\_Property.type->forAll(t|not((t.name='IDREFS')and(t.\_'package'.name='XMLPrimitiveTypes')))

###### NDR3 [Rule 9-53] (REF,EXT). No attribute type of xs:ENTITY

[Rule 9-53](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-53), No attribute type of xs:ENTITY (REF, EXT): [Section 9.2.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3.1), Prohibited attribute types

**[OCL] context** XSDProperty **inv:**

(self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)   
implies   
self.base\_Property.type->forAll(t|not((t.name='ENTITY')and(t.\_'package'.name='XMLPrimitiveTypes')) )

###### NDR3 [Rule 9-54] (REF,EXT). No attribute type of xs:ENTITIES

[Rule 9-54](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-54), No attribute type of xs:ENTITIES (REF, EXT): [Section 9.2.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3.1), Prohibited attribute types

**[OCL] context** XSDProperty **inv:**

(self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)   
implies   
self.base\_Property.type->forAll(t|not((t.name='ENTITIES')and(t.\_'package'.name='XMLPrimitiveTypes')))

###### NDR3 [Rule 9-55] (REF,EXT). No attribute type of xs:anySimpleType

[Rule 9-55](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-55), No attribute type of xs:anySimpleType (REF, EXT): [Section 9.2.3.1](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3.1), Prohibited attribute types

**[OCL] context** XSDProperty **inv:**

(self.kind=NIEM\_UML\_Profile::NIEM\_PSM\_Profile::XSDPropertyKindCode::attribute)   
implies   
self.base\_Property.type->forAll(t|not((t.name='anySimpleType')and(t.\_'package'.name='XMLPrimitiveTypes')))

###### NDR3 [Rule 9-56] (REF,EXT). No attribute default values

[Rule 9-56](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-56), No attribute default values (REF, EXT): [Section 9.2.3.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3.2), No attribute value constraints

[English]

This constraint enforced by provisioning, there are no @default attributes generated for an xs:attribute within a target InformationModel schema.

###### NDR3 [Rule 9-57] (REF,EXT). No attribute fixed values

[Rule 9-57](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#rule_9-57), No attribute fixed values (REF, EXT): [Section 9.2.3.2](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.2.3.2), No attribute value constraints

[English]

This constraint enforced by provisioning, there are no @fixed attributes generated for an xs:attribute within a target InformationModel schema.

###### XSDProperty AttributeKind

If kind=attribute, then an XSDProperty must have multiplicity 1..1 or 0..1, must not be a derived union and must not subset any other property. If the type is not empty, it must be a DataType.

**[OCL] context** XSDProperty **inv:**

(self.kind.oclAsType(EnumerationLiteral).name='attribute')   
implies(  
 (self.base\_Property.upper=1)  
 and not (self.base\_Property.isDerivedUnion)  
 and self.base\_Property.subsettedProperty->isEmpty()  
 and( not(self.base\_Property.type.oclIsUndefined()) implies self.base\_Property.type.oclIsKindOf(DataType) )  
 )

###### XSDProperty Owner

An XSDProperty must be the ownedAttribute of a DataType or a Class stereotyped as a NIEMType.

**[OCL] context** XSDProperty **inv:**

self.base\_Property.namespace.namespace.stereotypedBy('InformationModel')

### <Stereotype> XSDRepresentationRestriction

##### Description

XSDRepresentationRestriction specifies a restriction on the representation in an XML schema of the values of a base DataType.

##### Extends

UML::DataType

##### Properties

###### whiteSpace : XSDWhiteSpaceCode [0..1]

whiteSpace is a restriction on the value space of the DataType. It is implemented in XML Schema as the value of the value attribute on the xsd:whiteSpace element, the child of the xsd:restriction element which is the immediate child of the xsd:simpleType element.

##### Constraints

###### must have one generalization

A DataType with an XSDRepresentationRestriction must have exactly one

generalization.

**[OCL] context** XSDRepresentationRestriction **inv:**

self.base\_DataType.generalization->size()=1

### <Stereotype> XSDSimpleContent

##### Description

The «XSDSimpleContent» stereotype represents a relationship between two type definitions: the first is a complex type definition with simple content, the second is a simple type.

If the complex type definition is a «Restriction» of another complex type definition with simple content, then the simple type defines the constraining facets of the xsd:restriction to the other complex type. Otherwise, the relationship is implemented in XML Schema through base attribute on the xsd:extension element of the first type definition, the actual value of which resolves to the second type definition.

Section 3.4 of [XML Schema Structures](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/) addresses simple content types in XML Schema; [Section 9.1.3.3](http://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/NIEM-NDR-3.0-2014-07-31.html#section_9.1.3.3) of [NIEM-NDR] addresses simple content types in NIEM-conformant XML Schema.

##### Extends

UML::Realization

##### Constraints

###### Client must be a «NIEMType»

The client of an XSDSimpleContent Realization must be a Class stereotyped as a NIEMType.

**[OCL] context** XSDSimpleContent **inv:**

self.base\_Realization.client->forAll(client|client.oclIsKindOf(Classifier) and client.namespace.stereotypedBy('InformationModel'))

###### supplier must be a DataType

The suppler of an XSDSimpleContent Realization must be a DataType.

**[OCL] context** XSDSimpleContent **inv:**

self.base\_Realization.supplier->forAll(s|s.oclIsKindOf(DataType))

### <Enumeration> XSDProcessContentsCode

##### Description

XSDProcessContentsCode supports the processContents attribute of the XSDAnyProperty stereotype.

##### Literals

###### strict

###### lax

###### skip

### <Enumeration> XSDPropertyKindCode

##### Description

XSDPropertyKindCode supports the kind attribute of XSDProperty by providing values to specify if an XSD property is represented as an xsd:element or xsd:attribute.

##### Literals

###### element

###### attribute

### <Enumeration> XSDWhiteSpaceCode

##### Description

Enumeration XSDWhiteSpaceCode supports the whiteSpace attribute of the XSDWhiteSpaceCode attribute as per the XSD definitions.

##### Literals

###### replace

###### collapse

###### preserve

## Profile : NIEM\_UML\_Profile

### Overview

The NIEM UML Profile imports the NIEM PIM Profile, the NIEM PSM Profile and the Model Package Description Profile, so all three of these profiles can effectively be imported just by importing the single NIEM UML Profile.

# NIEM-UML Transformation Reference