Intelligence Community Technical Specification

XML Data Encoding Specification for Virtual Coverage

Version 2015-AUGr2017-JUL

July 21, 2017

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

1.1 - Purpose

This *XML Data Encoding Specification for Virtual Coverage* (VIRT.XML) defines detailed implementation guidance for using Extensible Markup Language (XML) to encode virtual coverage data. This Data Encoding Specification (DES) defines the XML elements and attributes, associated structures and relationships, mandatory and cardinality requirements, and permissible values for representing VIRT data concepts using XML.

1.2 - Scope

This specification is applicable to the Intelligence Community (IC) and information produced by, stored, or shared within the IC. This DES may have relevance outside the scope of intelligence; however, prior to applying outside of this defined scope, the DES should be closely scrutinized and differences separately documented and assessed for applicability.

1.3 - Background

The Intelligence Community Chief Information Officer (IC CIO) is leading the IC's enterprise transformation to an "interoperable federated architecture." Intelligence Community Directive (ICD) 500, *Director of National Intelligence Chief Information Officer* [7] grants the IC CIO the authority and responsibility to:

- Develop an Intelligence Community Enterprise Architecture (IC EA).
- Lead the IC's identification, selection, development, and management of IC enterprise standards.
- Incorporate technically sound, de-conflicted, interoperable enterprise standards into the IC EA.
- Certify that IC elements adhere to the architecture and standards.

In the area of enterprise standardization, the IC CIO is called upon to establish common Information Technology (IT) standards, protocols, and interfaces, to establish uniform information security standards, and to ensure information technology infrastructure, enterprise architecture, systems, standards, protocols, and interfaces support the overall information sharing strategies and policies of the IC as established in relevant law, policy, and directives.

Enterprise standards facilitate the information exchanges, service protocols, network configurations, computing environments, and business processes necessary for a service-enabled federated enterprise. As the enterprise develops and deploys shared services employing approved standards, not only will information and services be interoperable, but significant efficiencies and savings will be achieved by promoting capability reuse. As detailed in Intelligence Community Standard (ICS) 500-21, *Tagging of Intelligence and Intelligence-Related Information* [10] the extensive and consistent use of Extensible Markup Language (XML) within data encoding specifications allows for improved data exchanges and processing of information, thereby facilitating achievement of the IC's data discovery, data sharing, and interoperability goals.

An encoding specification defines a concrete implementation – a file format for example – for concepts in the *IC Abstract Data Definition* [2]. Many IC encoding specifications are based on XML,

but other technologies are possible. For example, IC-ID[4] defines a plain-text format for IC Identifiers as well as an associated XML structure.

1.4 - Enterprise Need

Information sharing within the national intelligence enterprise will increasingly rely on describing virtual locations in shared intelligence. A structured, verifiable representation of virtual coverage to the intelligence data is required in order for the enterprise to become inherently "smarter" about the information flowing in and around it. Such a representation, when implemented with other data formats, improved user interfaces, and data processing utilities, can provide part of a larger, robust information assurance infrastructure capable of automating some of the management and exchange decisions today being performed by human beings.

The IC has standardized the various classification and control markings established for information sharing of portions within the Information Security Markings (ISM) specification of the Intelligence Community Enterprise Architecture (ICEA) Data Standards. The Virtual Coverage XML specification uses the ISM specification to facilitate portion marking of virtual coverage needs.

Enterprise needs and requirements for this specification can be found in the following Office of the Director of National Intelligence (ODNI) policies and implementation guidance:

- IC Information Technology Enterprise (IC ITE):
  - Intelligence Community Information Technology Enterprise (IC ITE) Increment 1 Implementation Plan[3]
- 500 Series:
  - Intelligence Community Directive (ICD) 500, Director Of National Intelligence Chief Information Officer[7]
  - Intelligence Community Standard (ICS) 500-21, Tagging of Intelligence and Intelligence-Related Information[10]
- 200 Series:
  - Intelligence Community Directive (ICD) 208, Write for Maximum Utility[5]
  - Intelligence Community Directive (ICD) 209, Tearline Production and Dissemination[6]

1.5 - Audience and Applicability

DESs are primarily intended to be used by those developing tools and services to create, modify, store, exchange, search, display, or further process the type of data being described.

The conditions of use and applicability of this technical specification are defined outside of this technical specification. IC Standard (ICS) 500-20, *Intelligence Community Enterprise Standards Compliance*,[9] defines the IC Enterprise Standards Baseline (IC ESB) and the applicability of such to an IC element.

The IC ESB defines the compliance requirements associated with each version of a technical specification. Each version will be individually registered in the IC ESB. The IC ESB will define, among other things, the location(s) of the relevant artifacts, prescriptive status, and validity period, all of which characterize the version and its utility.

Additional applicability and guidance may be defined in separate IC policy guidance.

1.6 - Conventions

Certain technical and presentation conventions were used in the creation of this document to ensure readability and understanding.

1.6.1 - Language

When appearing in all capital letters in this technical specification, the keywords “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” are to be interpreted as described in IETF RFC 2119, “Key words for use in RFCs to Indicate Requirement Levels.” [11] When these words appear in regular case, they are meant in their natural-language sense.

1.6.2 - Typography

Certain typography is used throughout the body of this document to convey certain meanings, in particular:

- *Italics* – A title of a referenced work or a specialized or emphasized term
- Underscore – An abstract data element
- **Bold** – An XML element or attribute

1.6.3 - Terminology

For an implementation to conform to this specification, it MUST adhere to all normative aspects of the specification. For the purposes of this document, normative and informative are defined as:

- *Normative*: considered to be prescriptive and necessary to conform to the standard.
- *Informative*: serving to instruct, enlighten or inform.

1.6.4 - XML Namespaces

Namespaces referenced in this document and the prefixes used to represent them are listed in the following table. The namespace prefix of any XML Qualified Name used in any example in this document should be interpreted using the information below.

Table 1 - XML Namepaces

Prefix	URI
ism	urn:us:gov:ic:ism

1.7 - Dependencies

1.7.1 - Types of Dependencies

Specifications often rely on other specifications, components or artifacts, either directly or indirectly. Dependencies play an important role in functionality or provide informational

relationships between the various artifacts. The following terms are defined to help assist with understanding how the various artifacts work together:

Dependency	Directly or transitively influenced by. Examples: 1. A is influenced by B therefore B is a dependency of A. 2. A is influenced by B and B is influenced by C; therefore C is a dependency of A.
Direct Dependency	Explicit influence. Example: A influences B.
Inverse Dependency	Directly or transitively influences. Example: B influences A.

1.7.2 - Specification Dependencies

This technical specification directly depends on the technical specifications, documentation, and implementations listed in Table 2. The dependencies listed below are directly referenced in this specification (e.g. Schema, Schematron), and are normative or informative as indicated.

The subsequent figure, Figure 1, is an informative graphical representation of all of the IC CIO specifications related to this specification. The graphic depicts direct dependencies (see Direct Dependency). However, the representations may not match an exact schema import tree or dependency diagram that an analysis of the Schema, Schematron or other documents would yield. For example, the graphic only shows a given specification once even though it may actually be imported by many specifications or be a direct dependency. All specifications listed in Table 2 will be shown in Figure 1; however not all specifications listed in Figure 1 may appear in Table 2. Figure 1 is to aid users in gaining a general understanding of all direct dependencies.

Table 2 - Dependencies

Name	Dependency Description
XML Data Encoding Specification for Information Security Marking Metadata (ISM.XML.V13+)[12]	The specification does not depend on a specific version of Information Security Marking Metadata (ISM.XML); ISM.XML versions later than version 13 MAY be used. The minimum version was based on the earliest non-retired version; ESB 17-1 was used for determining the version.

Name	Dependency Description
Schematron[15]	Schematron — ISO/IEC 19757-3:2006 — is a rule-based document schema definition language. In this specification Schematron is a formal language used to express normative business rules, so this reference is normative. The Schematron rules are normative in the sense that they convey criteria that a document MUST adhere to, exactly as English may be used to convey normative criteria. It is not necessary for implementers to use the specific Schematron encoding in this specification. Implementers MAY use any encodings, tools, or languages desired to implement validation schemes for conformance to this specification. Note: The Schematron rules in this specification use XSLT 2.0[21] query binding.
XSLT 2.0[21] implementation of Schematron[15] by Rick Jelliffe (2010-04-14) Note: The only available identifying descriptors for this implementation are the implementer's name and date of release. This implementation may be found at the following URL: http://code.google.com/p/schematron/.	The International Organization for Standardization does not create nor endorse reference implementations of its standards. For the purposes of this specification the *behavior* of the implementation created by Mr. Jelliffe is normative. Implementers MAY use any encodings, tools, or languages desired to implement validation schemes for conformance to this specification. To conform to this specification, a validator MUST find a document valid *if and only if* the Schematron implementation by Mr. Jelliffe would find the document valid according to the Schematron rules in this specification.
Value enumerations used for several XML structures are defined in the various Controlled Vocabulary Enumerations included in this DES.	Specification uses CVEs to encode controlled vocabularies. The use of the VIRT CVEs is normative.

Figure 1 : Related Specifications

1.7.3 - Standalone and Convenience Packages

The standalone package of this specification does not include the specifications that it is dependent on since there may be more recent versions of those specifications available. There is a convenience package of the specification that includes the most recent versions of all direct dependent (see Direct Dependency) specifications at the time the package is generated. It is anticipated that this convenience package will be updated when any of the dependent specifications change; however, it will not be signed as a formal package. In order to obtain all the necessary standalone packages, this specification's dependencies and their dependencies will have to be traversed and obtained. These packages will have to be downloaded and copied into the appropriate directories for paths to the schema and controlled vocabulary enumerations (CVEs) to validate and operate as intended.

Convenience packages convey all dependencies pre-packaged together and are tested as interoperable. When trying to mix and match versions that have not been pre-packaged together, there may be risk that a particular combination may not be compatible, especially when mixing with versions of specifications that were not available at the time of a specification's release.

1.7.4 - Inverse Dependencies

Generally, it is only necessary to think of the *direct dependencies* (see Direct Dependency) in the dependency tree. However, with the specification versions being decoupled, it is also important to consider the *inverse dependencies* (see Inverse Dependency), for compatibility with newer versions of a given specification. The changes introduced to a given specification can sometimes make it incompatible with current versions of its inverse dependencies.

Since this specification is one such specification that is used by other specifications released by the IC CIO, the Figure 2 has been included to assist readers in understanding all of the dependency relationships and how changes in a specification may impact others. This diagram is representative of dependencies at the time of the release of this specification, but are subject to change over time.

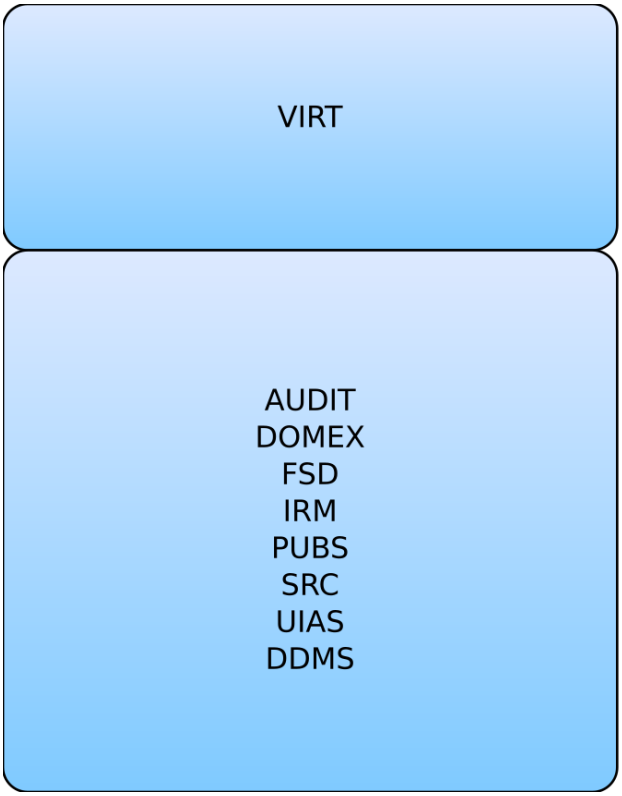

Figure 2 : Inverse Dependency Specifications

1.8 - Conformance

The XML schemas (unless noted otherwise), CVE values from the XML CVE files, and any Schematron[15] rules are normative for this specification. The rest of this document and the rest of this package, including the descriptive content referenced within the XML Schema Guide, the XSL transformations, the SchematronGuide, and PDF CVE value files, are informative. Additionally, the use of keywords defined in IETF RFC 2119[11] is considered normative within the scope of the sentence. All other parts of this document are informative.

The XML schemas provided may import other specifications. The versions of dependency specifications imported are not normative in that to import a different version of a component specification you could modify the import or substitute a different version of the component using the existing import path. This could be done by changing the schema file or by using XML Catalogs.[19] For example, a schema could be changed to incorporate a different version of a dependency like ISM by changing the attribute declaration of **@ism:DESVersion**='9' to **@ism:DESVersion**='10' in the xsd:schema statement. The ability to specify which version of a dependent specification to import enables the configuration change control of parent specifications (such as PUBS and TDF) to be "decoupled" from the configuration change control of dependent specifications (such as ISM CVE updates). This "decoupling" method has not been in place for all versions of these parent specifications; therefore, please verify with the dependency table to ensure use of allowed dependency versions.

Additional guidance that is either classified or has handling controls can be found in separate annexes distributed to the appropriate networks and environments as necessary. Systems and services operating in those environments MUST consult the appropriate annexes.

1.9 - Version Policies

1.9.1 - XML Namespace Policy

The XML namespaces defined in this specification do not incorporate a version number and do not change with revisions of the specification. This choice aligns with perspective two from “The Disposition of Names in an XML Namespace.”[16] This decision allows for systems that process information encoded with these specifications to use the same XPath expressions across multiple revisions. It was agreed the burden of updating all XPath based systems for every revision to the specification was unacceptable. See section 4.2.2 “Versioning and XML namespace policy” of “Architecture of the World Wide Web, Volume One.”[17]

There is a version attribute (e.g. **@DESVersion**, **@CESVersion**, **@TESVersion**, **@version**) for each namespace defined in an IC CIO specification. Version attributes are used to capture the specification version number the specification author intends an instance to conform to. Namespaces do not change, so the version attribute is required to fully understand an instance document.

As changes to the specification are released, the version number captured in the “version” attribute increments. See Section 1.9.2 - Version Numbering for information on the numbering scheme.

This XML namespace policy only applies to the namespaces defined in this specification, any namespaces that are included by reference should define their own namespace policy.

1.9.2 - Version Numbering

The version numbering for this specification is defined by a year-month structure (e.g., YYYY-MMM). This provides a temporal representation of when the specification was released. Revisions to a version of the specification also use a year-month structure (e.g., YYYY-MMM). When the version number is used in the version attribute, the expression follows the Augmented Backus–Naur Form[1] below:

Version Format when used in the version attribute:

```
[1]          Version : := Year Month["." Revision ] ["-" CustomizationSuffix ]
[2]    VersionYear : := 4( DIGIT )
[3]  VersionMonth : := 2( DIGIT )
[4] Customization : := 1*23(ALPHA / DIGIT / "_" )
                Suffix
[5]  RevisionYear : := 4( DIGIT )
[6]  RevisionMont : := 2( DIGIT )
                   h
[7]        Revision : := Year Month
```

Version in XML Lexicon

The following vocabulary helps explain the meaning of terms used in the version documentation, and it may further constrain the set of allowable values:

Version	The version number as it might be expressed in a DESVersion, CESVersion or other XML attribute for indicating the version/revision being referenced.
VersionYear	The four digit year from the version of the specification being referenced.
VersionMonth	The 2 digit month from the version of the specification being referenced.
CustomizationSuffix	An optional suffix used when customizing a version of a specification. This would be used to indicate that you have extended the specification in some fashion for a particular use case.
RevisionYear	The four digit year from the revision of the specification being referenced.
RevisionMonth	The 2 digit month from the revision of the specification being referenced.
Revision	The Year and Month from the revision of the specification being referenced. Revisions are modifications to Versions.

Chapter 2 - Development Guidance

2.1 - Relationship to Abstract Data Definition and other encodings

The relationship of the XML structures defined in this encoding specification to the abstract terms defined in the ADD are described using a mapping table in the ADD. The mapping tables generally show the mapping to the encoding specification where a structure is defined, not where it is used. These mappings are provided for reference only. The complete set of encoding specification artifacts, both normative and informative, should be consulted in order to gain a complete understanding of this encoding specification.

The mappings in the ADD provide a starting point for the development of automated transformations between formats defined by the encoding specifications. However, it should be noted that when these transformations are used between formats with different levels of detail there might be some data loss.

2.2 - Additional Guidance

2.2.1 - Usage of ISM

ISM attributes used in VIRT reflect the security markings required for access rights management and handling. However, VIRT is NOT a standalone specification; it does not include the necessary dependencies and structures to produce an instance document that is valid to the ISM specification. VIRT is a reusable component meant to be integrated into another specification that does fully implement ISM.

2.3 - CSV Notes

There are Comma Separated Value files provided for all of the CVEs. They are in the CVE folder with the XML and JSON versions of the information. They are provided to assist developers using the CVEs; the specifications do not use them at this moment. There are no new requirements because of their existence.

Important

The CSV files on many systems will open “automatically” in Microsoft Excel; the default opening however, will not correctly read UTF-8 special characters. These are found in some country names such as “Republic of Côte d’Ivoire”. If you need to use a CVE that contains such special characters, or you think may contain such characters in Excel, you should:

- Open Excel to a blank sheet
- Under the Data menu choose to get external data from a text file
- Choose UTF-8 as the file origin

- Choose delimited as the format
- Choose next
- Change from tab to Comma as the delimiter
- Finish import to get the data in with the UTF-8 Characters properly encoded in Excel.

2.4 - JSON Notes

There are JSON format files provided for all of the CVEs. They are in the CVE folder with the XML and CSV versions of the information. They are provided to assist developers using the CVEs; the specifications do not use them at this moment. There are no new requirements because of their existence. The JSON files are formatted using JSON-LD based on a proposed method for JSON in NIEM.

Chapter 3 - Definitions, Interfaces, and Constraints

3.1 - Constraint Rule Types

Data constraint rules fall into two categories - validation and rendering constraints. Data validation constraints explicitly define policy validation constraints, describing how data should be structured and encoded in order to comply with IC policy. Validation constraint rules are implemented as a combination of basic XML Schema constraints and supplemental constraints for more complex rules. Complex constraint rules contain technical rule descriptions, Schematron rule implementations, and *Human Readable* descriptions. The human readable text describes the intent and meaning behind the more technical rule description. The semantics of the constraint rules are normative, whereas the use of the Schematron implementation is informative. Implementers developing alternative validation code should follow the technical rule descriptions and Schematron logic. Should there be a perception of conflict, implementers should bring it to the attention of the appropriate configuration control body for resolution. Rendering constraint rules define constraints on the display and rendering of documents. While expressed in a similar manner to the data validation constraint rules, there is no expectation that evaluation of these rules can be automated; rather these rules should inform the evaluation of a system's capabilities and functionality.

3.2 - "Living" Constraint Rules

These constraint rules are a "living" rule set. The constraint rules provided are a valid starter set and do not attempt to address the full scope of business rules addressed by authoritative guidance. These rules will be expanded and modified as the model matures, and as applicable security marking policies change.

Since these constraint rules are only a subset of the entire rule base, an XML document that is compliant with these rules may still not be fully compliant with all of the business rules defined in the authoritative guidance. An XML document that is not compliant with these rules is not compliant with the authoritative guidance.

3.3 - Classified or Controlled Constraint Rules

Additional rules that are either classified or have handling controls can be found in separate annexes closely associated with the encoding specification artifacts wherever they are located.

3.4 - Constraint Terminology

For the purposes of this document, the following statements apply:

- The term "is specified" indicates that an attribute is applied to an element and the attribute has a non-null value.
- The term "must be specified" indicates that an attribute MUST be applied to an element and the attribute MUST have a non-null value.
- The term "is not specified" indicates that an attribute is not applied to an element, or an attribute is applied to an element and the attribute has a null value.

- The term “must not be specified” indicates that an attribute MUST NOT be applied to an element.

3.5 - Errors and Warnings

The severity of a constraint rule violation is categorized as either an “Error” or a “Warning.” An “Error” is more severe and is indicative of a clear violation of a constraint rule, which would be likely to have a significant impact on the quality of a document. A “Warning” is less severe although noteworthy, and may not necessarily have any impact on the quality of a document. The severity of a constraint rule violation is indicated in brackets preceding each constraint rule description.

Each system responsible for processing a document (e.g., create, modify, transform, or exchange) MUST make a mission-appropriate decision about using a document with errors or warnings based on mission needs.

3.6 - Rule Identifiers

Each constraint rule has an assigned rule identifier, indicated in brackets preceding the constraint rule description. VIRT.XML data validation constraint rule identifiers are prefixed with "VIRT-ID-" and followed by a 5 digit unique number, assigned from pre-defined ranges to group rules by classification. The numerical ranges are described in Table 3. As the constraint rules are managed over time, IDs from deleted rules will not be reused.

Table 3 - Numerical Rule Identifier Ranges

Rule Identifier Range		Description
Start	End	
00001	09999	Reserved for Unclassified constraint rules
10001	19999	Reserved for Unclassified but For Official Use Only (FOUO) constraint rules
20001	20999	Reserved for constraint rules classified at the “Secret//REL USA, FVEY” level
21001	21999	Reserved for constraint rules classified at the “Secret//NF” level
22001	29999	Reserved for constraint rules classified at the “Secret//TBD” level
30001 and above		Reserved for constraint rules classified with other classifications

3.7 - Data Validation Constraint Rules

3.7.1 - Purpose

The VIRT.XML schema defines the data elements, attributes, cardinalities and parent-child relationships for which XML instances must comply. Validation of these syntax aspects is an important first step in the validation process. An additional level of validation is needed to ensure that the content complies with the constraints as specified in applicable IC policy guidance and codified in these constraint rules. Traditional schema languages are generally unable to effectively represent these additional constraints.

3.7.2 - Schematron

Schematron[15] is the formal language used in this specification to encode normative data validation constraints. The Schematron rules are normative in the sense that they convey criteria a document MUST meet, exactly as English may be used to convey normative criteria.

It is not necessary for implementers to use the specific Schematron encoding in this specification, and implementers MAY use any encodings, tools, or languages desired to implement validation schemes for conformance to this specification. To conform to this specification, a validator MUST find a document valid *if and only if* the Schematron implementation by Mr. Jelliffe would find the document valid according to the Schematron rules in this specification.

For better understanding, the Schematron[15] rules for this specification may be executed in *Oxygen®* [14] or with an XSLT 2.0-compliant processor using the XSLT 2.0[21] transforms in the Schematron implementation from Rick Jelliffe (see XSLT 2.0 implementation of Schematron by Rick Jelliffe in the Dependency table).

The constraint rules for this specification are dependent on XPath 2.0[20] and XSLT 2.0[21] features. Regarding the use of XPath 2.0 and XSLT 2.0 with Schematron, the editor of the ISO Schematron standard stated the following:[13]

> By default, Schematron uses the XPath language as used in XSLT 1.0, and is typically implemented by converting the schema into an XSLT 1.0 script which is run against the document being validated. However, ISO Schematron also allows XSLT 2.0 to be used, and this is becoming an increasingly popular choice because of the extra expressive convenience of XPath 2.0: a different skeleton is available for this.

Note

For convenience, the specification package provides the XSLT 2.0[21] implementation of Schematron[15] along with a compiled version of the rules.

3.7.3 - Non-null Constraints

XML syntax allows all elements with content declared to be of data type “string” to have zero or more characters of content, meaning elements can be empty or null. According to this specification, all required elements (and certain conditional elements) MUST have content, other than white space.[1] Elements, which are allowed to only have text content, MUST have text content specified.

3.7.4 - Inherited Constraints

In an instance of VIRT.XML, the use of attributes and elements from supplementary data encoding specifications must be fully conformant with the constraint rules defined in those specifications. For a full list of supplementary specifications, see Section 1.7 - Dependencies.

[1] “White space” is defined in XML 1.0[18] as “(white space) consists of one or more space (#x20) characters, carriage returns, line feeds, or tabs.”

3.7.5 - Value Enumeration Constraints

Several elements and attributes of the VIRT.XML model use CVEs to define the data allowed in the element or attribute. In some cases the specific CVE is specified via an attribute, which may include a default CVE. Further, in some of the cases where the CVE can be specified, the attribute may restrict the list of CVEs allowed and some may allow for the author to specify their own CVE. For each of these, the value must be in the specified external CVE or the default CVE.

Some CVEs are not available on all networks. A subset CVE will be provided for use on networks not approved for the entire list. If the processing will occur on a network where the entire CVE is not available, the subset CVE may be substituted in the constraint rules since the excluded values would be excluded from use on the lower network.

As noted in the specific rules, a failure of validation against a CVE will generate an Error.

3.7.6 - Additional Constraints

3.7.6.1 - DES Constraints

The DES version is specified through attributes on the root element. The schema constrains the values of these attributes. The **DESVersion** attribute enables systems processing an instance document to be certain which set of constraint rules, schema, CVEs and business rules are intended by the author to be used.

3.7.6.2 - Revision Constraints

When validating an instance document against the validation rule sets and schema provided by the specification there is a certain philosophy that SHOULD be applied to both protect the data and the systems processing that data. This validation philosophy consists of the following seven basic rules that describe how the DESVersion matters to validation:

1. One MUST NOT validate with rules older than the integer version declared in an instance; this is an error.

2. One MAY validate with rules that are of a greater integer version than an instance.

3. When validating an instance with a lower integer version number than that of the validation rules, there MAY be a minimum integer version cutoff for a set of rules. If such a limit exists, this is an error.

4. Within an integer, validation MUST only occur with the newest decimal value implemented by the validator; that is a validator MUST only implement one signed validation rule set within an integer and it SHOULD be the latest.

5. When a validator detects an instance document claiming a version newer than what is implemented in the validator, a notice/log SHOULD be generated so a human can evaluate if the validator needs to be updated to the latest rule set, as passing the old rules MAY not comply with current law or policy.

6. A validator SHOULD document and communicate all versions and revisions it accepts, including the constraints (business/policy rules, allowed values, schema formats, etc.) in each of those versions.

The matrix of fictional generic examples in Table 4 are provided to illustrate these validation concepts with the following assumptions:

- Version 11: Technically incompatible with newer versions
- Version 12: Technically compatible with newer versions, but retired from the Enterprise Standards Baseline
- Version 13: Oldest in the Enterprise Standards Baseline
- Version 13.201701: Revision to version 13
- Version 13.201804: Revision to version 13
- Version 201508: Standard release
- Version 201609: Latest version release

Table 4 - Revision Constraints table

Validation Rules Version / Instance Version	11	12	13	13.201701	13.201804	201508	201609
11	Version Match	Instance Too Old (Tech)	Instance Too Old (Tech)	Instance Too Old (Tech)	Instance Too Old (Tech)	Instance Too Old (Tech)	Instance Too Old (Tech)
12	Instance Too New	Version Match	Instance Too Old (ESB)	Instance Too Old (ESB)	Instance Too Old (ESB)	Instance Too Old (ESB)	Instance Too Old (ESB)
13	Instance Too New	Instance Too New	Version Match	Same Integer	Same Integer	Allowed	Allowed
13.201701	Instance Too New	Instance Too New	Same Integer	Version Match	Same Integer	Allowed	Allowed
13.201804	Instance Too New	Instance Too New	Same Integer	Same Integer	Version Match	Allowed	Allowed
201508	Instance Too New	Instance Too New	Instance Too New	Instance Too New	Instance Too New	Version Match	Allowed
201609	Instance Too New	Instance Too New	Instance Too New	Instance Too New	Instance Too New	Instance Too New	Version Match

3.7.7 - Constraint Rules

The detailed constraint rules for the VIRT.XML schema can be found in a separate document inside the SchematronGuide directory, in the VIRT_Rules.pdf file. This document is generated from the individual Schematron files to provide a single searchable document for all of the constraint rules encoded in Schematron. Obsolete rule numbers are listed in the SchematronGuide.

3.8 - Data Rendering Constraint Rules

3.8.1 - Purpose

Rendering rules define constraints on the rendering and display of VIRT.XML documents. The intent is to inform the development of systems capable of rendering or displaying VIRT.XML data for use by individuals not familiar with the details of the VIRT.XML markup. While expressed in a similar manner to the data validation constraint rules above, there is no expectation that evaluation of these rules can be automated; rather these rules should inform the evaluation of a system's capabilities and functionality.

3.8.2 - Rendering Constraint Rules

The following table contains the information for the VIRT.XML data rendering constraint rules.

Table 5 - Constraint Rules

Rule Number	Severity	Description	Human Readable Description
There are no Data Rendering Constraint rules at this time.			

Chapter 4 - Conformance Validation

An instance document conforms with this specification if it conforms to all normative guidance of this specification and this specification's dependencies and it passes all of the following validation steps. This specification does not dictate how this validation strategy is implemented.

4.1 - Schema Validation

An instance document MUST comply with the schemas for this specification and this specification's dependencies, and schema validation SHOULD occur prior to other validation steps. If schema validation fails, results from later steps may be indeterminate.

4.2 - Business Rule Validation

An instance document MUST comply with the business rules expressed in this specification and those expressed in this specification's dependencies. The business rules in this specification are expressed in Schematron, but it is not necessary for implementers to use the specific Schematron encoding in this specification. Implementers MAY use any encodings, tools, or languages desired to implement validation schemes for conformance to this specification. To conform to this specification, a validator MUST find a document valid *if and only if* the Schematron implementation by Mr. Jelliffe would find the document valid according to the Schematron rules in this specification.

Chapter 5 - Generated Guides

5.1 - Schema Guide

The detailed description and reference documentation for the VIRT.XML schema can be found as a collection of HTML files inside the SchemaGuide directory. These files comprise a guide that serves as an interactive presentation of the VIRT.XML schema as well as an implementation-specific data element dictionary.

The guide was generated with a commercially available product named *oXygen®*, produced by SyncRO Soft.

The guide provides an interactive index to:

- Global Elements and Attributes
- Local Elements and Attributes
- Simple and Complex Types
- Groups and Attribute Groups
- Referenced Schemas

Where applicable, the guide provides:

- Diagram
- Namespace
- Type
- Children (Child Elements)
- Used by
- Properties
- Patterns
- Enumerations
- Attributes
- Annotations
- Source Code

The guide is published in a folder consisting of the master HTML file *SchemaGuide.html* with supporting graphics.

5.2 - Schematron Guide

The detailed description and reference documentation for the VIRT.XML Schematron rules can be found in a separate document named *VIRT_Rules.pdf*, which is located inside the SchematronGuide directory. This document is generated from the individual Schematron files to provide a single searchable document for all of the constraint rules encoded in Schematron.

Appendix A Feature Summary

The following table shows the version dependencies for VIRT on other specifications. Direct dependencies are marked with an asterisk.

Table 6 - VIRT Dependency over Time

Dependent DES	V1	V2015-AUG	V2015-AUGr2017-JUL
ISM*	V9+	V9+	V13+
NTK	V7+	N/A	N/A
ISMCAT			V2016-SEP+

The following table summarizes major features by version for this VIRT and all dependent specs.

Table 7 - Feature Summary Legend

Key	Description
F	Full (able to comply and verified by spec to some degree)
P	Partial (Able to comply but not verifiable)
N	Non-compliance (Can’t comply)
N/A	Not Applicable. Feature is no longer required.
Cell Colors represent the same information as the Key value	

A.1. VIRT Feature Summary

Table 8 - VIRT Feature Comparison

VIRT Feature Comparison				
Required date	Feature	V1	V2015-AUG	V2015-AUGr2017-JUL
	Supports multiple versions of ISM.XML (V9 - Current)	F	F	F
	Support for NTK attribute based controls.	F	N/A	N/A
	Addition of new networks.	N	F	F

Appendix B Change History

The following table summarizes the version identifier history for this DES.

Table 9 - DES Version Identifier History

Version	Date	Purpose
1	21 January 2013	Initial Release
2015-AUG	13 August 2015	Added missing networks values to CVE
2015-AUGr2017-JUL	21 July 2017	Routine revision to technical specification. For details of changes, see Section B.1 - 2015-AUGr2017-JUL Change Summary

B.1 - 2015-AUGr2017-JUL Change Summary

Significant drivers for Version 2015-AUGr2017-JUL include:

- Community Change Requests

Table 10 summarizes the changes made to this technical specification from Version 2015-AUG to revision 2015-AUGr2017-JUL.

Table 10 - V2015-AUGr2017-JUL Change History

#	Change	Artifacts Changed	Compatibility Notes
1	Added the following networks: BICES, CFBLNet, DDTE in support of DI2E to VIRT Network (CR-2016-045)	CVEs CVEnum-VIRTNetworkName.xml modified	Data generation and ingestion systems need to be updated to accommodate the new values.
2	Create JSON version of CVEs in VIRT (CR-2017-070)	CVEs	No impact to systems.
3	Create CSV version of CVEs in VIRT (CR-2017-049)	CVEs	No impact to systems.
4	Added DESVersion enforcement rule as warning (CR-2017-098)	Schema Schematron VIRT-ID-00003 added VIRT_XML.sch modified	Data generation and ingestion systems need to be updated to accommodate the changes to the rules.
5	Added inverse dependency section and definitions for Dependencies and Inverse Dependencies. (CR-2017-127)	Documentation	No impact to systems.

#	Change	Artifacts Changed	Compatibility Notes
6	The schema change logs will no longer be maintained as of the 2015-AUGr2017-JUL release. The existing change logs will only serve as legacy information. For changes to schema as of and after 2015-AUGr2017-JUL, reference the change history in the DES.	Schema	No impact to systems.
7	Added the revision constraint section since this is the first revision of VIRT.	Documentation	Data generation and ingestion systems will may need to be updated to properly validate against the right revisions of specifications.
8	Added DES section describing versioning strategy. (CR-2017-201)	Documentation	No impact to systems.
9	Added @id and @role to all sch:rule elements, in support of commercial tools warnings and errors and to support open source unit testing frameworks. (CR-2017-216)	All non-abstract Schematron rules modified	No impact to existing systems. Additional capabilities.
10	Update prose to align with current specifications. Specifically, change e-mail address to ic-standads-support@iarpa.gov, update dependency table to standardize wording. (CR-2017-235)	Documentation	No impact to systems.
11	Modified cardinality rendering. (CR-2017-023)	CVEs	No impact to existing systems, documentation rendering change only.
12	Update the version numbering EBNF to reflect the existence of Revisions. (CR-2017-237, CR-2017-260)	Documentation	No impact to systems.

B.2 - V2015-AUG Change Summary

Significant drivers for Version 2015-AUG include:

- Community Change Requests

Table 11 summarizes the changes made to this technical specification from Version 1 to Version 2015-AUG.

Table 11 - V2015-AUG Change History

Change	Artifacts Changed	Compatibility Notes
Added network values to CVE	CVE	Added network values
Updated code descriptions to improve readability.	Schematron	No impact to data generation and ingestion systems.
Removed NTK attribute based access	DES	Systems may not longer use the NTK attribute based access and will have to use the element based on instead.
Removed Dependency on NTK	DES Schema	Systems no longer require NTK to use VIRT
Deleted Schematron rule 00002	VIRT_ID_00002.sch	Rule duplicated a schema constraint, so there is no impact to implementing systems. Systems may be updated to remove the deleted rule at the discretion of the system's manager.

Appendix C List of Abbreviations

This appendix lists all the acronyms and abbreviations referenced in this encoding specification.

ADD	Abstract Data Definition
CVE	Controlled Vocabulary Enumeration
DES	Data Encoding Specification
DNI	Director of National Intelligence
ESB	Enterprise Standards Baseline
FOUO	For Official Use Only
HTML	HyperText Markup Language
IC	Intelligence Community
IC CIO	Intelligence Community Chief Information Officer
IC EA	Intelligence Community Enterprise Architecture
IC ESB	Intelligence Community Enterprise Standards Baseline
IC ITE	Intelligence Community Information Technology Enterprise
ICD	Intelligence Community Directive
ICS	Intelligence Community Standard
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
ISM	Information Security Markings
ISMCAT	Information Security Marking Country Codes and Tetragraphs
ISO	International Organization for Standardization
IT	Information Technology
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
NIEM	National Information Exchange Model
NTK	Need-To-Know Metadata
OCIO	Office of the Intelligence Community Chief Information Officer

ODNI	Office of the Director of National Intelligence
PUBS	Intelligence Publications
RFC	Request for Comments
TDF	Trusted Data Format
URL	Uniform Resource Locator
VIRT	Virtual Coverage
XML	Extensible Markup Language
XPath	XML Path Language
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations

Appendix D Bibliography

Bibliography

[1] ABNF
Internet Engineering Task Force. *Augmented BNF for Syntax Specifications: ABNF*.
Available online at: http://tools.ietf.org/html/std68
Also known as: http://www.ietf.org/rfc/rfc5234.txt

[2] ADD
Office of the Director of National Intelligence. *Intelligence Community Abstract Data Definition (IC-ADD.XML)*.
Available online Intelink-TS at: http://go.ic.gov/soSC6M8
Available online Intelink-U at: https://w3id.org/ic/standards/ADD
Available online at: https://w3id.org/ic/standards/public

[3] IC ITE INC1 IMPL
Office of the Director of National Intelligence. *Intelligence Community Information Technology Enterprise (IC ITE) Increment 1 Implementation Plan*. July 2012.
Available online Intelink-TS at: http://go.ic.gov/4X6TOc1

[4] IC-ID.XML
Office of the Director of National Intelligence. *Text and XML Data Encoding Specification for Intelligence Community Identifier (IC-ID.XML)*.
Available online Intelink-TS at: http://go.ic.gov/mQ4IUDk
Available online Intelink-U at: https://w3id.org/ic/standards/IC-ID
Available online at: https://w3id.org/ic/standards/public

[5] ICD 208
Office of the Director of National Intelligence. *Write For Maximum Utility*. Intelligence Community Directive 208. 17 December 2008.
Available online at: http://www.dni.gov/files/documents/ICD/icd_208.pdf

[6] ICD 209
Office of the Director of National Intelligence. *Tearline Production and Dissemination*. Intelligence Community Directive 209. 6 September 2012.
Available online at: http://www.dni.gov/files/documents/ICD/ICD%20209%20Tearline%20Production%20and%20Dissemination.pdf

[7] ICD 500
Office of the Director of National Intelligence. *Director of National Intelligence Chief Information Officer*. Intelligence Community Directive 500. 7 August 2008.
Available online Intelink-TS at: http://go.ic.gov/5Ot5sbK
Available online at: http://www.dni.gov/files/documents/ICD/ICD_500.pdf

[8] ICPG 710.1
Director of National Intelligence. *Application of Dissemination Controls: Originator Control*. Intelligence Community Policy Guidance 710.1. 25 July 2012.
Available online Intelink-TS at: http://go.ic.gov/0d147Ee

Available online at: http://www.dni.gov/files/documents/ICPG/ICPG710.1.pdf

[9] ICS 500-20
Director of National Intelligence Chief Information Officer. *Intelligence Community Enterprise Standards Compliance*. Intelligence Community Standard 500-20. 16 December 2010.
Available online Intelink-TS at: http://go.ic.gov/sLKNq3N
Available online Intelink-U at: https://w3id.org/ic/standards/policy/ICS500-20

[10] ICS 500-21
Director of National Intelligence Chief Information Officer. *Tagging of Intelligence and Intelligence-Related Information*. Intelligence Community Standard 500-21. 28 January 2011.
Available online Intelink-TS at: http://go.ic.gov/cWyv9nw
Available online Intelink-U at: https://w3id.org/ic/standards/policy/ICS500-21

[11] IETF-RFC 2119
Internet Engineering Task Force. *Key words for use in RFCs to Indicate Requirement Levels*. March 1997.
Available online at: http://tools.ietf.org/html/rfc2119

[12] ISM.XML
Office of the Director of National Intelligence. *XML Data Encoding Specification for Information Security Markings (ISM.XML)*.
Available online Intelink-TS at: http://go.ic.gov/3oipfOY
Available online Intelink-U at: https://w3id.org/ic/standards/ISM
Available online at: https://w3id.org/ic/standards/public

[13] Jelliffe
Richard Alan Jelliffe. *FAQ*. Frequently Asked Questions: Is Schematron tied to XSLT 1.0?.
Available online at: http://www.schematron.com

[14] Oxygen
SyncRO Soft. *<oXygen/> XML Editor*.
Available online at: http://www.oxygenxml.com/

[15] Schematron
International Organization for Standardization (ISO). *Information technology -- Document Schema Definition Language (DSDL) -- Part 3: Rule-based validation -- Schematron*. ISO/IEC 19757-3:2006.
ISO Spec Available online at: http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html
StyleSheets for compiling Available online at: http://code.google.com/p/schematron/

[16] TAG-9-Jan-2006
W3C Technical Architecture Group (TAG). *The Disposition of Names in an XML Namespace*. 9 January 2006.
Available online at: http://www.w3.org/2001/tag/doc/namespaceState.html

[17] WEBARCH-15-Dec-2004
W3C. *Architecture of the World Wide Web, Volume One*. 15 December 2004.

Available online at: http://www.w3.org/TR/webarch

[18] XML 1.0
World Wide Web Consortium (W3C). *Extensible Markup Language (XML) 1.0, Second Edition*. W3C, 6 October 2000.
Available online at: http://www.w3.org/TR/2000/REC-xml-20001006

[19] XML Catalogs
The Organization for the Advancement of Structured Information Standards [OASIS]. *XML Catalogs*. Committee Specification 06 Aug 2001.
Available online at: https://www.oasis-open.org/committees/entity/spec-2001-08-06.html

[20] XPath2
World Wide Web Consortium (W3C). *XML Path Language (XPath) 2.0 (Second Edition)*. W3C Recommendation 14 December 2010 (Link errors corrected 3 January 2011).
Available online at: http://www.w3.org/TR/xpath20/

[21] XSLT2
World Wide Web Consortium (W3C). *XSL Transformations (XSLT) Version 2.0*. W3C Recommendation 23 January 2007.
Available online at: http://www.w3.org/TR/xslt20/

Appendix E Points of Contact

The Intelligence Community Chief Information Officer (IC CIO) facilitates one or more collaboration and coordination forums charged with the adoption, modification, development, and governance of IC technical specifications of common concern. This technical specification was produced by the IC CIO and coordinated with these forums, approved by the IC CIO or a designated representative, and made available at the following DNI-sponsored web sites.

Public Website: https://w3id.org/ic/standards/public

Intelshare: https://w3id.org/ic/standards/data-specs

Direct all inquiries about this IC technical specification, IC technical specification collaboration and coordination forums, or IC element representatives involved in those forums, to the IC CIO.

E-mail: ic-standards-support@iarpa.gov.

Appendix F IC CIO Approval Memo

An Office of the Intelligence Community Chief Information Officer (OCIO) Approval Memo should accompany this enterprise technical data specification bearing the signature of the Intelligence Community Chief Information Officer (IC CIO) or an IC CIO-designated official(s). If an OCIO Approval Memo is not accompanying this specification's version release package, then refer back to the authoritative web location(s) for this specification to see if a more complete package or a specification update is available.

Specification artifacts display a date representing the last time a version's artifacts as a whole were modified. This date most often represents the conclusion of the IC Element collaboration and coordination process. Once the IC Element coordination process is complete, the specification goes through an internal OCIO staffing and coordination process leading to signature of the OCIO Approval Memo. The signature date of the OCIO Approval Memo will be later than the last modified date shown on the specification artifacts by an indeterminable time period.

Upon signature of the OCIO Approval Memo, IC Elements may begin to use this specification version in order to address mission and business objectives. However, it is critical for IC Elements, prior to disseminating information encoded with this new specification version, to ensure that key enterprise services and consumers are prepared to accept this information. IC Elements should work with enterprise service providers and consumers to orchestrate an orderly implementation transition to this specification version in concert with mandatory and retirement usage decisions captured in the IC Enterprise Standards Baseline as defined in Intelligence Community Standard (ICS) 500-20.[9]