



Intelligence Community Technical Specification

XML Data Encoding Specification for Trusted Data Format

Version 2014-DEC-r2017-JUL

July 21, 2017

Distribution Notice:

This document has been approved for Public Release and is available for use without restriction.

Table of Contents

Chapter 1 - Introduction	1
1.1 - Purpose	1
1.2 - Scope	1
1.3 - Background	1
1.4 - Enterprise Need	2
1.5 - Audience and Applicability	2
1.6 - Conventions	3
1.6.1 - Language	3
1.6.2 - Typography	3
1.6.3 - Terminology	3
1.7 - Dependencies	3
1.7.1 - Types of Dependencies	3
1.7.2 - Specification Dependencies	4
1.7.3 - Standalone and Convenience Packages	7
1.7.4 - Inverse Dependencies	8
1.8 - Conformance	8
1.9 - Version Policies	9
1.9.1 - XML Namespace Policy	9
1.9.2 - Version Numbering	9
Chapter 2 - Development Guidance	11
2.1 - Relationship to Abstract Data Definition and other encodings	11
2.2 - TDF Structure	11
2.2.1 - Version Declarations	14
2.3 - Assertions	15
2.3.1 - Assertion Scopes	15
2.3.1.1 - Assertion Scopes Within TDO	15
2.3.1.2 - Assertion Scopes Within TDC	15
2.3.1.3 - HandlingAssertion scopes within TDO	17
2.3.1.4 - HandlingAssertion scopes within TDC	17
2.3.2 - Mission-Specific Metadata Assertions	17
2.3.3 - Assertions and Data State	17
2.4 - Binding and BindingInfo	18
2.5 - Normalization Method	21
2.6 - Encryption and EncryptionInfo	22
2.7 - Linked or Embedded Data Objects	22
2.8 - MIME type	22
2.9 - CSV Notes	23
2.10 - JSON Notes	23
Chapter 3 - Definitions, Interfaces, and Constraints	24
3.1 - Constraint Rule Types	24
3.2 - "Living" Constraint Rules	24
3.3 - Classified or Controlled Constraint Rules	24
3.4 - Constraint Terminology	24
3.5 - Errors and Warnings	25
3.6 - Rule Identifiers	25
3.7 - Data Validation Constraint Rules	25

3.7.1 - Purpose	25
3.7.2 - Schematron	26
3.7.3 - Non-null Constraints	26
3.7.4 - Inherited Constraints	26
3.7.5 - Value Enumeration Constraints	27
3.7.6 - Additional Constraints	27
3.7.6.1 - DES Constraints	27
3.7.6.2 - Revision Constraints	27
3.7.7 - Constraint Rules	29
3.8 - Data Rendering Constraint Rules	29
3.8.1 - Purpose	29
3.8.2 - Rendering Constraint Rules	29
Chapter 4 - Conformance Validation	30
4.1 - Definitions	30
4.2 - Why a verbose validation strategy is required	30
4.3 - How to determine the ISM version within structured content	31
4.4 - Required Order of Handling Assertions	32
4.5 - TDO Validation Steps	32
4.5.1 - Step 1 - TDO aware and cross assertion constraints	32
4.5.2 - Step 2 - Extension point constraints	33
4.5.3 - Step 3 - TDO structure constraints	33
4.5.4 - Step 4 - ISM consistency constraints	34
4.5.4.1 - Step 4a - Consistency constraints for Assertions with resource level portion markings	34
4.5.4.2 - Step 4b - Consistency constraints for Payloads with resource level portion marking	35
4.5.4.3 - Step 4c - Consistency constraints for Assertions and Payloads with non-resource level markings	35
4.6 - TDC Validation Steps	36
4.6.1 - Step 1 - TDC aware and cross assertion constraints	36
4.6.2 - Step 2 - Extension point constraints	36
4.6.3 - Step 3 - TDC structure constraints	37
4.6.4 - Step 4 - ISM consistency constraints	37
4.6.4.1 - Step 4a - Consistency constraints for Assertions with resource level portion markings	37
4.6.4.2 - Step 4b - Consistency constraints for Assertions with non-resource level markings	37
4.6.5 - Step 5 - Recursive Validation	38
Chapter 5 - Generated Guides	39
5.1 - Schema Guide	39
5.2 - Schematron Guide	40
Chapter 6 - Future Features	41
6.1 - Explicit Scope	41
6.2 - BoundValueList	41
Appendix A - Feature Summary	42
A.1 - IC-TDF Feature Summary	42
Appendix B - Change History	44
B.1 - V2014-DEC-r2017-JUL Change Summary	44
B.2 - V2014-DEC Change Summary	50

B.3 - V3 Change Summary	51
B.4 - V2 Change Summary	52
Appendix C - List of Abbreviations	54
Appendix D - Bibliography	56
Appendix E - Points of Contact	60
Appendix F - IC CIO Approval Memo	61

List of Figures

Figure 1 - Related Specifications	7
Figure 2 - Inverse Dependency Specifications	8
Figure 3 - Simple TDO	12
Figure 4 - TDO with Encryption	12
Figure 5 - TDF Structure	13
Figure 6 - TDF Detailed Structure	14
Figure 7 - Trusted Data Collection (TDC)	14
Figure 8 - TDF Extension Points	31

List of Tables

Table 1 - Dependencies	4
Table 2 - TDO Binding Contents	19
Table 3 - TDC Binding Contents	20
Table 4 - Sample URLs for XML Canonicalization Normalization Methods	22
Table 5 - Numerical Rule Identifier Ranges	25
Table 6 - Revision Constraints table	28
Table 7 - Constraint Rules	29
Table 8 - TDF Dependency over Time	42
Table 9 - Feature Summary Legend	42
Table 10 - IC-TDF Feature comparison	42
Table 11 - DES Version Identifier History	44
Table 12 - Data Encoding Specification V2014-DEC-r2017-JUL Change Summary	44
Table 13 - Data Encoding Specification V2014-DEC Change Summary	50
Table 14 - Data Encoding Specification V3 Change Summary	51
Table 15 - Data Encoding Specification V2 Change Summary	52

Chapter 1 - Introduction

1.1 - Purpose

This *XML Data Encoding Specification for Trusted Data Format* (IC-TDF.XML) defines detailed implementation guidance for using Extensible Markup Language (XML) to encode IC-TDF 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 trusted data format 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* ^[9] 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* ^[14] 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^[6] 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 information assurance metadata (including enterprise data headers) to allow interagency access control, automated exchanges, and appropriate protection of shared intelligence. A structured, verifiable representation of security metadata bound 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 within the Information Security Markings (ISM), Need-To-Know (NTK), Information Resource Metadata (IRM), Enterprise Data Header (EDH), and Access Rights and Handling (ARH) XML specifications of the Intelligence Community Enterprise Architecture (IC EA) Data Standards. The IC Trusted Data Format XML specification further expands on this body of work, adapting and extending it as necessary for TDF to function as the IC submission format for binding assertion metadata with data resource(s). This TDF functionality supports the IC way ahead strategy of implementing secure cloud-based information exchange and discovery on the IC Enterprise.

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^[4]
- 500 Series:
 - Intelligence Community Directive (ICD) 500, Director Of National Intelligence Chief Information Officer^[9]
 - Intelligence Community Directive (ICD) 501, Discovery and Dissemination or Retrieval of Information within the IC^[10]
 - Intelligence Community Standard (ICS) 500-21, Tagging of Intelligence and Intelligence-Related Information^[14]
- 200 Series:
 - Intelligence Community Directive (ICD) 208, Write for Maximum Utility^[7]
 - Intelligence Community Directive (ICD) 209, Tearline Production and Dissemination^[8]
 - Intelligence Community Policy Memorandum (ICPM) 2007-200-2, Preparing Intelligence to Meet the Intelligence Community's Responsibility to Provide^[12]

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*, [13] defines the IC Enterprise Standards Baseline (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.” [15] 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.

1.7 - 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 1](#). 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 1](#) will be shown in [Figure 1](#); however not all specifications listed in [Figure 1](#) may appear in [Table 1](#). [Figure 1](#) is to aid users in gaining a general understanding of all direct dependencies.

Table 1 - Dependencies

Name	Dependency Description
<i>XML Data Encoding Specification for Information Security Marking Metadata</i> (ISM.XML.V13+) ^[16]	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
<i>XML Data Encoding Specification for Need-To-Know Metadata</i> (NTK.XML.V10+) ^[18]	The specification does not depend on a specific version of Need To Know (NTK.XML); NTK.XML versions later than version 10 MAY be used. The minimum version was based on the earliest non-retired version; ESB 17-1 was used for determining the version.
<i>XML Data Encoding Specification for Enterprise Data Header</i> (IC-EDH.XML.V4+) ^[5]	This specification does not depend on a specific version of Enterprise Data Header (IC-EDH.XML); IC-EDH.XML versions later than version 4 MAY be used. The minimum version was based on the earliest non-retired version; ESB 17-1 was used for determining the version.
<i>XML Data Encoding Specification for Access Rights and Handling</i> (ARH.XML.V3+) ^[3]	This specification does not depend on a specific version of Access Rights and Handling (ARH.XML); ARH.XML versions later than version 3 MAY be used. The minimum version was based on the earliest non-retired version; ESB 17-1 was used for determining the version.
<i>XML Data Encoding Specification for Revision Recall</i> (RevRecall.XML.V2014-DEC+) ^[21]	This specification does not depend on a specific version of Revision Recall (RevRecall.XML); RevRecall.XML versions later than version 2014-DEC MAY be used. The minimum version was based on a technical dependency; The promotion from regular assertion to handling assertion.
Schematron ^[22]	<p>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.</p> <p>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.</p> <p>Note: The Schematron rules in this specification use XSLT 2.0^[28] query binding.</p>

Name	Dependency Description
<p>XSLT 2.0^[28] implementation of Schematron^[22] by Rick Jelliffe (2010-04-14)</p> <p>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/.</p>	<p>The International Organization for Standardization does not create nor endorse reference implementations of its standards. For the purposes of this specification the <i>behavior</i> of the implementation created by Mr. Jelliffe is normative.</p> <p>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 <i>if and only if</i> the Schematron implementation by Mr. Jelliffe would find the document valid according to the Schematron rules in this specification.</p>
<p>Value enumerations used for several XML structures are defined in the various Controlled Vocabulary Enumerations (CVEs) included in this DES.</p>	<p>Specification uses CVEs to encode controlled vocabularies. The use of the IC-TDF CVEs is normative.</p>

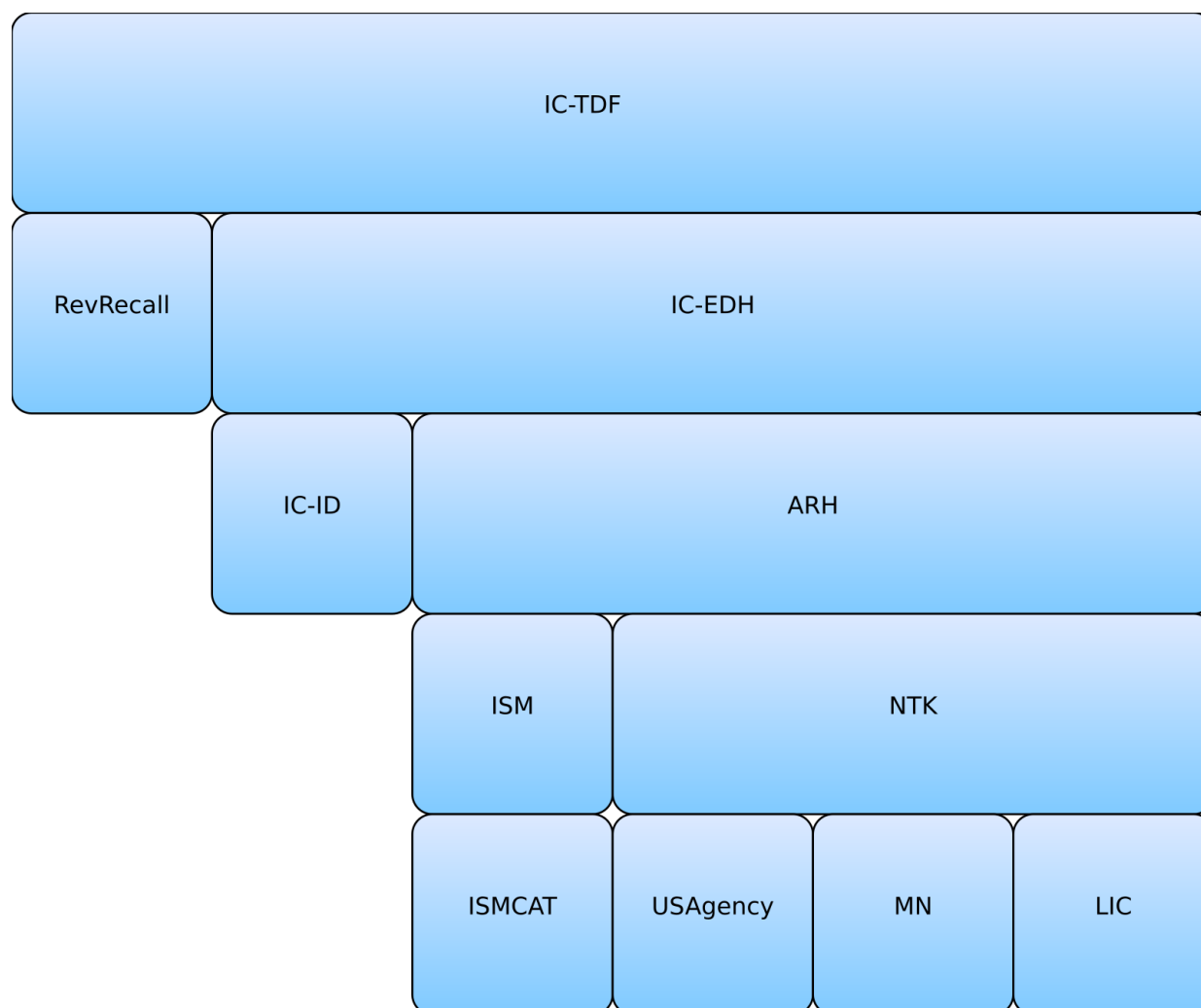


Figure 1 : Related Specifications

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.

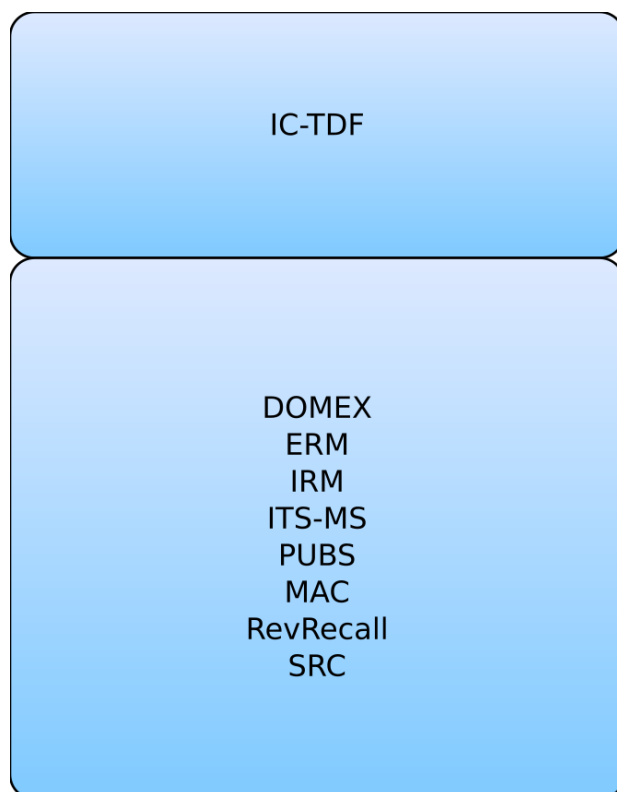


Figure 2 : Inverse Dependency Specifications

1.8 - Conformance

The XML schemas (unless noted otherwise), CVE values from the XML CVE files, and any Schematron^[22] 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^[15] 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.^[26] 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."^[23] 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."^[24]

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 - TDF Structure

The IC-TDF.XML specification has a consistent and simple concept of Assertions and Payloads. There are two options for root elements: TrustedDataObject (TDO) and TrustedDataCollection (TDC). A TDO contains some data (the payload) and some statements about that data (the assertions). In the context of TDF, an 'assertion' is defined as a statement providing handling, discovery, or mission metadata describing a payload, TDO, or TDC, depending on the scope of the assertion. To facilitate handling and access control decisions, each TDO and TDC must contain at least one HandlingAssertion. A HandlingAssertion is a special type of structured assertion that cannot be encrypted. In general it contains the IC Enterprise Data Header IC-EDH for the TDO or payload, providing the attributes needed for policy decisions regarding access control and how the data must be handled. ISM and NTK markings are contained in Handling Assertions, as part of the Access Rights and Handling block. In addition to the IC-EDH, there MAY also be an optional RevisionRecall HandlingAssertion. Additional discovery and mission assertions may also be provided as standard Assertions. A TDC contains a list of TDOs (the payload) and some statements about those TDOs (the assertions). A TDC may also be a collection of collections, and contain other TDCs.

Each TDO consists of one or more assertions and a payload. Assertions may optionally be cryptographically bound to the payload to provide assurance over the integrity of the assertion, the payload, and the relationship between the assertion and payload.

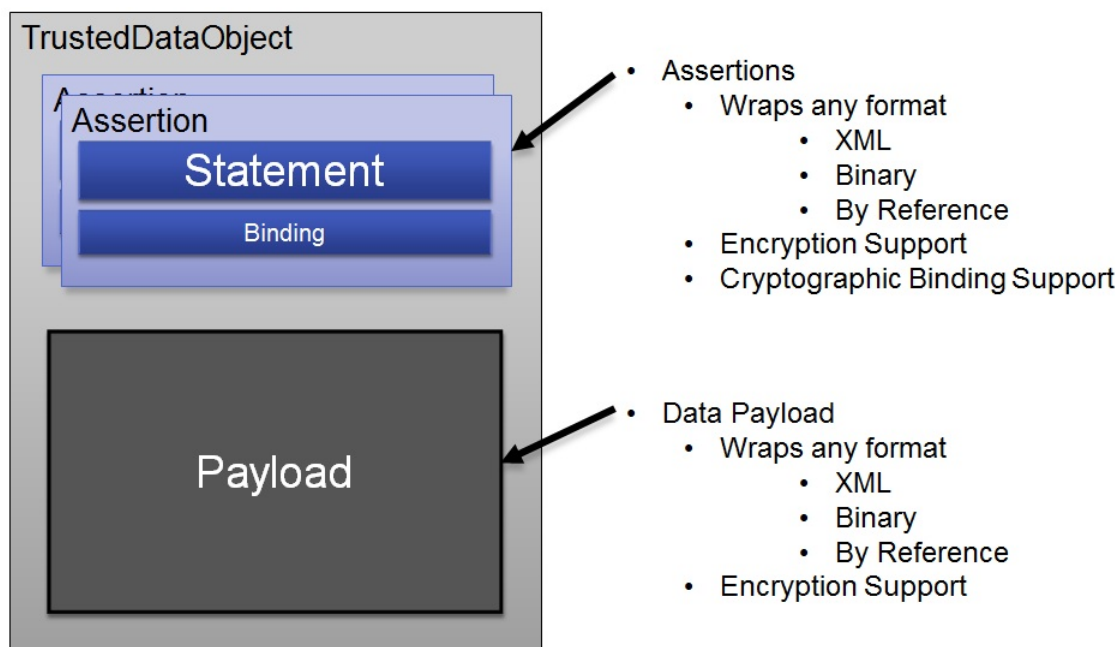


Figure 3 : Simple TDO

In a scenario where encryption is required, the TDO assertion statements and/or TDO payload may be optionally encrypted:

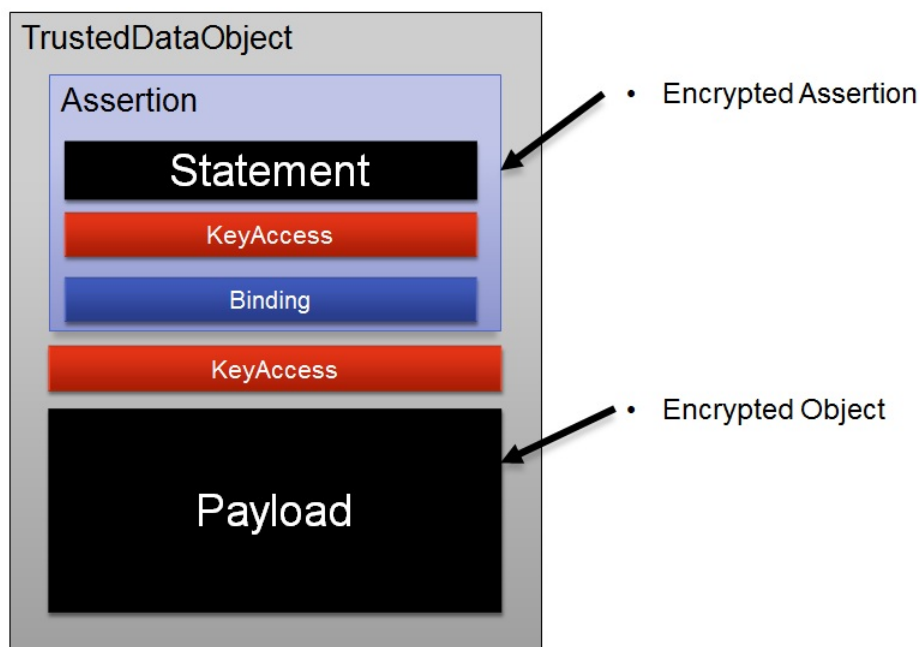


Figure 4 : TDO with Encryption

Each IC-TDF requires at least one IC-EDH handling assertion, optional Revision Recall handling assertion, optional discovery and mission assertions, and a payload. The handling assertion must

consist of a structured IC-EDH block. A common discovery assertion might be a structured IRM block. Mission specific metadata may consist of a structured block (XML) or unstructured data (binary). The payload may be structured XML, unstructured data, or a reference.

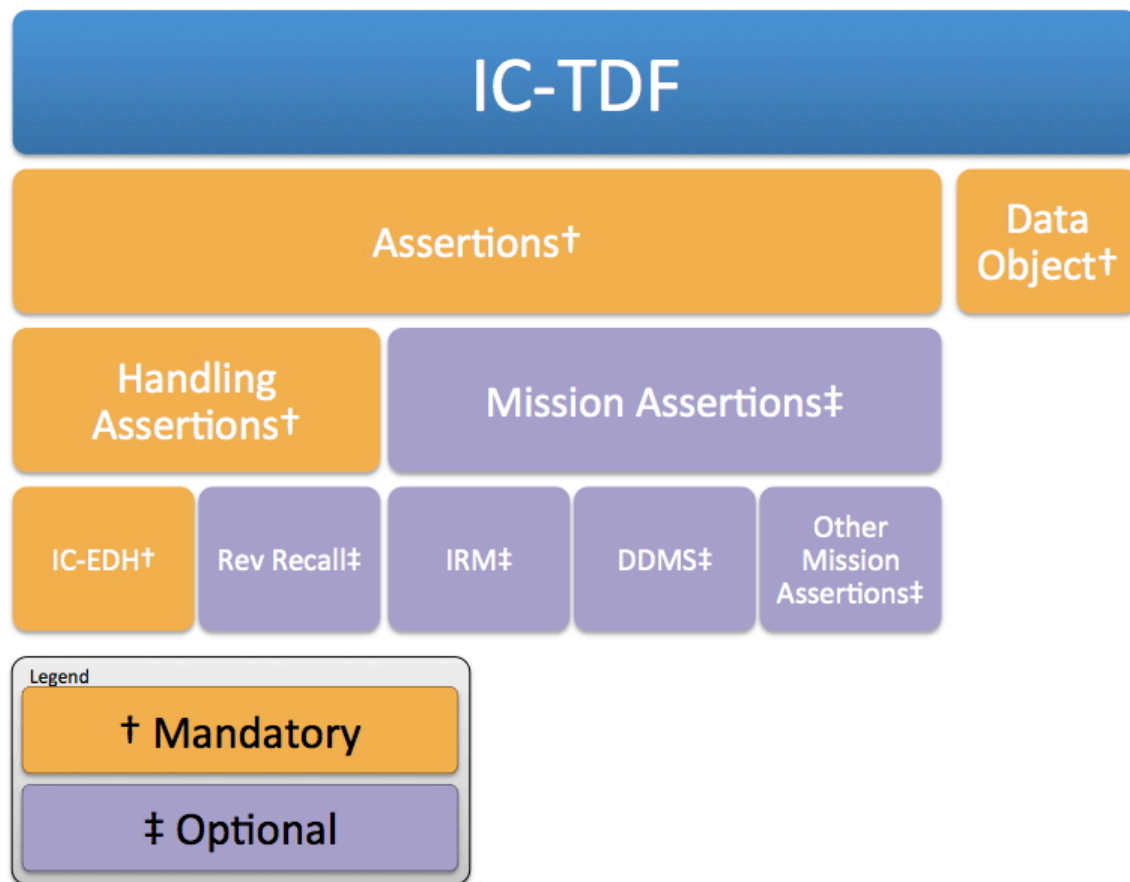


Figure 5 : TDF Structure

The diagram below shows expected use of IC specifications within a TDO. The use of the IC-EDH handling assertion and payload are required, whereas the discovery and mission specific assertions are optional.

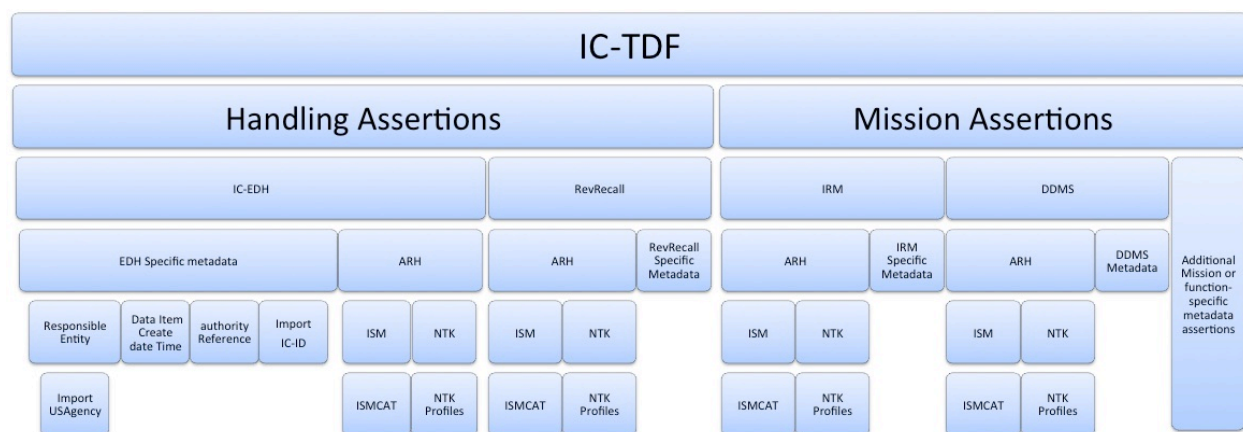


Figure 6 : TDF Detailed Structure

A TDC consists of a collection of TDOs or TDCs. It is expected but not required that the child TDOs and TDCs within a TDC are in some way related, with relationships encoded in the TDC assertions. For example, in a biometric use case, a TDC might correspond to a biometric identity, with child TDOs corresponding to biometric modalities, such as finger prints, iris scans, and facial images. In this biometric use case the root TDC assertions would describe the entire identity, while the child TDO assertions would describe the individual modalities.

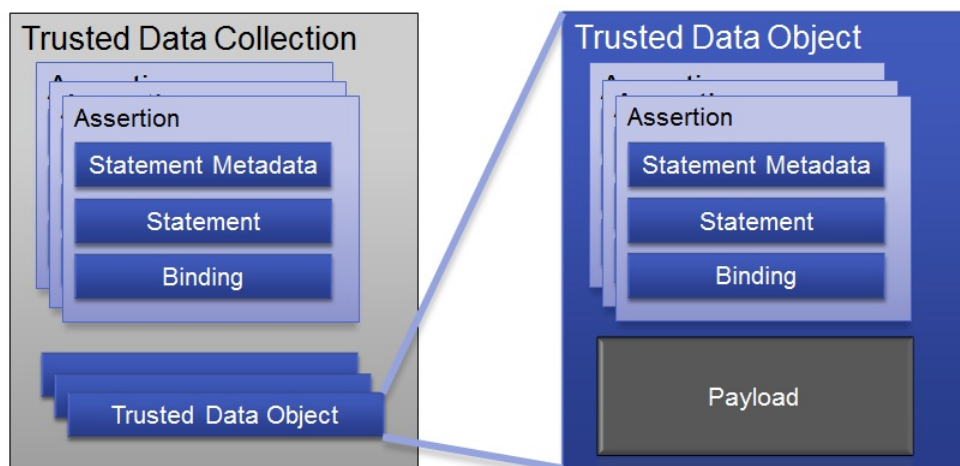


Figure 7 : Trusted Data Collection (TDC)

2.2.1 - Version Declarations

Specification versions are generally declared at the highest level of the XML structure that makes sense for its usage, generally either the root, or the first level of element that uses a specification.

As such, many specifications used in a TDF are generally declared at the root (i.e. ISM, NTK, IC-EDH, etc.).

As extension points, assertions and payloads may have different versions of specifications specified for use inside itself. For example, the TDF may declare an ISM DESVersion of [201412] while the payload might be a legacy document that declares the ISM DESVersion to be [9]. In this case that payload would be validated with ISM.XML.v9.

However, it is also possible that an assertion or payload does not contain declarations for versions of specifications. In this case they are considered to be the same versions that are declared in the TDF. That is, the extension points inherit specification versions from the TDF in which they reside and, if they are extracted from the TDF, those version declarations **MUST** be copied into that content during extraction to maintain validity as well as comprehensibility.

2.3 - Assertions

2.3.1 - Assertion Scopes

Assertions can be scoped to apply to different portions of an IC-TDF instance. Several assertion scopes imply certain meaning and processing instructions. The following sections explain the valid assertion scopes for use within TDOs and TDCs and any additional processing requirements they imply.

2.3.1.1 - Assertion Scopes Within TDO

Assertions within a TDO can be scoped to apply to either to the entire TDO, the payload only, or both. The following tokens are used to specify the scope of assertions within a TDO:

1. [PAYL] means this assertion applies only to the payload within this TDO.
2. [TDO] means this assertion applies to every element within the TDO other than itself (includes peer HandlingAssertions, Assertions, and the Payload). This scope essentially means "the entire TDO".

2.3.1.2 - Assertion Scopes Within TDC

Assertions within a TDC can be scoped to apply to several different portions of a TDC instance. [Definition: The child TDOs and TDCs contained within a TDC are referred to as the *collection members*.]

[Definition: An assertion with a *transitive scope* recursively applies to specific portions of each collection member within this TDC and MAY be inherited by a collection member if that collection member is extracted from this TDC.] Each transitive scope defines exactly which portions of the collection members the assertion applies to and how the assertion must be inherited when a collection member is extracted. Transitive scopes help reduce the need for duplicate assertions within collection members. For example, instead of making an identical assertion in each collection member individually, a single assertion with a transitive scope at the TDC level may have the same intent with much less overhead.

[Definition: An assertion with a *non-transitive* scope does not recursively apply to each collection member within this TDC and MUST NOT get inherited by a collection member if that collection member is extracted from this TDC.] Each non-transitive scope defines exactly which portion of this TDC the assertion applies to. Non-transitive scopes are used for assertions which only have meaning when considered in the scope of the TDC.

Whenever any change is made to the TDC, the intent of an assertion may no longer logically apply depending upon the assertion's scope and the change that was made. If a collection member is removed from the TDC, then the intent of an assertion with a transitive scope still logically applies to the remaining subset of collection members. However, any other change made to the collection members within the TDC may logically invalidate an assertion with a transitive scope (e.g., a new collection item is added or an existing collection member is modified). The intent of an assertion with a non-transitive scope may no longer logically apply if any modification is made to the portions of the TDC to which the assertion applies. Users modifying the TDC should understand the intent of each existing assertion in order to correctly preserve their intent or make some corrective modification after changes have been made. [Section 2.4 - Binding and BindingInfo](#) outlines how to cryptographically bind an assertion to the portions of the document to which it applies.

The following list defines the tokens used to specify the scope of assertions within TDCs:

1. [TDC] is a non-transitive scope and means this assertion applies to all TDC elements collectively (other than itself). This includes peer HandlingAssertions, Assertions, TrustedDataObjects, and TrustedDataCollections. This scope essentially means "the entire TDC".
2. [DESC_TDO] (short for descendant TDO) is a transitive scope and means this assertion applies to every TDO contained within this TDC.

When a collection member is extracted from this TDC it MAY inherit assertions with scope [DESC_TDO] from its ancestor TDCs in the following ways:

If the collection member being extracted is a TDO, then any assertion with scope [DESC_TDO] in an ancestor TDC becomes an assertion with scope [TDO] in the extracted TDO.

If the collection member being extracted is a TDC, then any assertion with scope [DESC_TDO] in an ancestor TDC becomes an assertion with scope [DESC_TDO] in the extracted TDC.

3. [DESC_PAYL] (short for descendant payload) is a transitive scope and means this assertion applies to every Payload within this TDC. This scope is similar to [DESC_TDO], but this scope applies ONLY to the Payloads within descendent TDOs and does NOT include any assertions or handling assertion of those TDOs.

When a collection member is extracted from this TDC it MAY inherit assertions with scope [DESC_PAYL] from its ancestor TDCs in the following ways:

If the collection member being extracted is a TDO, then any assertion with scope [DESC_PAYL] in an ancestor TDC becomes an assertion with scope [PAYL] in the extracted TDO.

If the collection member being extracted is a TDC, then any assertion with scope [DESC_PAYL] in an ancestor TDC becomes an assertion with scope [DESC_PAYL] in the extracted TDC.

4. [TDC_MEMBER] is a non-transitive scope and means this assertion applies to all collection members within this TDC. Unlike scope [TDC], this scope does not apply to peer HandlingAssertions and Assertions.

This scope is useful for making an assertion about the "current state" of the collection members within the TDC. For example, one might use the [TDC_MEMBER] scope to make an assertion that all members of the TDC contain biometric modalities for a certain individual. However, as soon as any modification is made to the collection members, then the assertion may no longer apply to the new state of the collection members (a collection member is added to the TDC, a collection member is removed from the TDC, any modification is made to any existing collection member).

2.3.1.3 - HandlingAssertion scopes within TDO

A TDO has at a minimum two HandlingAssertions: a *TDO handling assertion* and a *payload handling assertion*. This allows for separate access control decisions to be made for the payload versus the entire TDO (which includes the payload metadata). There may be an additional HandlingAssertion with scope [TDO] that contains Revision/Recall information using the RevRecall.XML^[21] specification. A HandlingAssertion MUST not be encrypted.

2.3.1.4 - HandlingAssertion scopes within TDC

A TDC can only have a single HandlingAssertion containing an IC-EDH^[5] specification and its scope must be [TDC]. There may also be an optional second HandlingAssertion scope [TDC] that contains Revision/Recall information for the TDC using the RevRecall.XML^[21] specification. A HandlingAssertion MUST not be encrypted.

2.3.2 - Mission-Specific Metadata Assertions

Although missions may create their own unique set of assertions, no understanding by the enterprise beyond access control is assured. The Assertion @type is intended to provide additional context allowing various systems to pre-determine relevance of assertions without parsing or reading all of the assertions. Assertion @type might include categorizations such as 'discovery,' 'mission,' or 'task order' to allow various systems to determine which assertions are relevant for them to parse.

2.3.3 - Assertions and Data State

If an assertion statement or a payload is encrypted, then there are in fact two (potentially different) markings needed for decision making, analysis, and querying: one for describing the handling required for the ciphertext and the other for the handling required for the unencrypted (and in effect external) state. In cases where statements and/or payloads are encrypted, handling assertions and statement metadata elements indicate whether their marks apply to the ciphertext vs. plaintext by using the attribute @tdf:appliesToState. This attribute may be leveraged in use cases such as:

- A user or system knows that they are not allowed to have/process data with NTK *systemXYZ*, and the user/system wants to query a large IC cloud repository and filter out results that require *systemXYZ* handling. For results with encrypted payloads, if the handling assertion only reflects the ciphertext handling (say Confidential) the user/system could get back thousands of encrypted results they cannot decrypt, should not see, and do not want to sort.
- Agency X publishes data to the IC cloud with encrypted payloads. In a decrypted state, the payload requires NTK markings that IC cloud cannot yet handle access-wise. In this case, when the markings in an assertion apply to state 'encrypted,' they should be part of rollup and used for the handling of the TDO. When the markings in an assertion apply to state 'unencrypted' they should be excluded from rollup, and used for search filtering, or access and processing decisions in systems that are able to decrypt the payload.

The attribute @tdf:appliesToState can be used with tdf:Assertion/tdf:StatementMetadata or with tdf:HandlingAssertion. The appliesToState attribute can only be used when content is encrypted, as indicated by the attribute @tdf:isEncrypted. When payload content is encrypted (@tdf:isEncrypted='true'), it must be marked with two HandlingAssertion blocks, one indicating the classification and handling required for the cyphertext payload (with @appliesToState='encrypted'), and the other indicating the classification and handling required for the plaintext payload after decryption (with appliesToState='unencrypted'). In this case, the HandlingAssertion that applies to the plaintext state is considered external to rollup, since the plain text content is not included in the instance. The appliesToState attribute should only be used with HandlingAssertions scoped to the payload. When Assertion statement content is encrypted (@tdf:isEncrypted='true') it must be marked with two StatementMetadata blocks: one indicating the classification and handling required to protect the cyphertext statement (with @tdf:appliesToState='encrypted') and the other indicating the classification and handling required to protect the plaintext statement after decryption (with @tdf:appliesToState='unencrypted'). In this case, the StatementMetadata describing the plaintext statement is considered external to rollup, since the plain text content is not included in the instance.

2.4 - Binding and BindingInfo

A key concept in the TDF specification is the ability to cryptographically assure the relationship among portions of the document. This assurance is represented by the optional **Binding** element available on each Assertion and HandlingAssertion.

The **Binding** element includes information about the algorithm used to calculate the signature, the **SignatureValue**.

In the current version of IC-TDF the **SignatureValue** is always calculated over a concatenation of the normalized portions of the document in the same order they appear in the document described by the Assertion.

The normalization method expressed in **Binding/SignatureValue/@normalizationMethod** is a URI that provides guidance on how to format the included values such as whitespace, attributes, and child nodes in a universally consistent manner. The normalization method is essential to prevent formatting such as white-space and order from interfering with the validation of the cryptographic integrity of data. For example, XML canonicalization is one form of normalization that might be utilized. More information on XML canonicalization is available online at: [W3C Canonical XML](http://www.w3.org/TR/xml-c14n) [http://www.w3.org/TR/xml-c14n]. To use XML canonicalization as a normalization

method, provide the URI to the form of XML canonicalization you are using, such as <http://www.w3.org/TR/2001/REC-xml-c14n-20010315> [http://www.w3.org/TR/2001/REC-xml-c14n-20010315] as the value for the **Binding/SignatureValue/@normalizationMethod**. This example URL is the URL defined in XML-SEC Rec for inclusive c14n without comments.

The expected portions of the document that each scope MUST include in the **SignatureValue** are detailed in the tables below. The abbreviation IFF stands for "if and only if". The pseudo XPath in the tables below are not syntactically valid and use some abbreviations to save space and improve readability:

Assume each element and attribute is in the IC-TDF namespace

Payload refers to the *TDF extension points* tdf:StringPayload, tdf:StructuredPayload, tdf:ReferenceValuePayload, and tdf:Base64BinaryPayload

AssertionStatement refers to the *TDF extension points* tdf:StringStatement, tdf:StructuredStatement, tdf:ReferenceStatement, and tdf:Base64BinaryStatement

HandlingStatement refers to an IC-EDH instance (Edh or ExternalEdh)

Table 2 - TDO Binding Contents

XPath	Required to include in binding
TrustedDataObject/ Assertion[@scope='PAYL']	<ol style="list-style-type: none"> 1. ./AssertionStatement 2. ./StatementMetadata IFF ./Binding/SignatureValue/@includesStatementMetadata='true' 3. ../Payload
TrustedDataObject/ HandlingAssertion[@scope='PAYL']	<ol style="list-style-type: none"> 1. ./HandlingStatement 2. ../Payload
TrustedDataObject/ Assertion[@scope='TDO'] or TrustedDataObject/ HandlingAssertion[@scope='TDO']	<ol style="list-style-type: none"> 1. ../HandlingAssertion/HandlingStatement 2. ../Assertion/AssertionStatement 3. ../Assertion/StatementMetadata IFF ./Binding/SignatureValue/@includesStatementMetadata='true' 4. ../Payload

Table 3 - TDC Binding Contents

XPath	Required to include in binding
TrustedDataCollection/ Assertion[@scope='TDC'] or TrustedDataCollection/ HandlingAssertion[@scope='TDC']	<ol style="list-style-type: none"> 1. ../HandlingAssertion/HandlingStatement 2. ../Assertion/AssertionStatement 3. ../Assertion/StatementMetadata IFF ../Binding/SignatureValue/ @includesStatementMetadata='true' 4. ../TrustedDataObject/HandlingAssertion/ HandlingStatement 5. ../TrustedDataObject/Assertion/ AssertionStatement 6. ../TrustedDataObject/Assertion/ StatementMetadata IFF ../Binding/ SignatureValue/ @includesStatementMetadata='true' 7. ../TrustedDataObject/Payload 8. ../TrustedDataCollection/ HandlingAssertion/HandlingStatement 9. ../TrustedDataCollection/Assertion/ AssertionStatement 10. ../TrustedDataCollection/Assertion/ StatementMetadata IFF ../Binding/ SignatureValue/ @includesStatementMetadata='true'

XPath	Required to include in binding
TrustedDataCollection/ Assertion[@scope='DESC_TDO'] or TrustedDataCollection/ Assertion[@scope='TDC_MEMBER']	<ol style="list-style-type: none"> 1. ./AssertionStatement 2. ./StatementMetadata IFF ./Binding/ SignatureValue/ @includesStatementMetadata='true' 3. ../TrustedDataObject/HandlingAssertion/ HandlingStatement 4. ../TrustedDataObject/Assertion/ AssertionStatement 5. ../TrustedDataObject/Assertion/ StatementMetadata IFF ./Binding/ SignatureValue/ @includesStatementMetadata='true' 6. ../TrustedDataObject/Payload 7. ../TrustedDataCollection/ HandlingAssertion/HandlingStatement 8. ../TrustedDataCollection/Assertion/ AssertionStatement 9. ../TrustedDataCollection/Assertion/ StatementMetadata IFF ./Binding/ SignatureValue/ @includesStatementMetadata='true'
TrustedDataCollection/ Assertion[@scope='DESC_PAYL']	<ol style="list-style-type: none"> 1. ./AssertionStatement 2. ./StatementMetadata IFF ./Binding/ SignatureValue/ @includesStatementMetadata='true' 3. ../TrustedDataObject/Payload

2.5 - Normalization Method

The normalization method expressed in Binding/SignatureValue/@normalizationMethod and Binding/BoundValueList/BoundValue/@normalizationMethod is a URI that provides guidance on how to format the included values such as whitespace, attributes, and child nodes in a universally consistent manner. The normalization method is essential to prevent formatting such as whitespace and order from interfering with the validation of the cryptographic integrity of data. For example, XML canonicalization is one form of normalization that might be utilized. The table below lists several XML canonicalization URLs.

Table 4 - Sample URLs for XML Canonicalization Normalization Methods

Sample NormalizationMethod URL	Description
http://www.w3.org/TR/2001/REC-xml-c14n-20010315	The URL defined in XML-SEC Rec for inclusive c14n without comments.
http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments	The URL defined in XML-SEC Rec for inclusive c14n with comments.
http://www.w3.org/2001/10/xml-exc-c14n#	The URL defined in XML-SEC Rec for exclusive c14n without comments.
http://www.w3.org/2001/10/xml-exc-c14n#WithComments	The URL defined in XML-SEC Rec for exclusive c14n without comments.
http://www.w3.org/2006/12/xml-c14n11	The URI for inclusive c14n 1.1 without comments.
http://www.w3.org/2006/12/xml-c14n11#WithComments	The URI for inclusive c14n 1.1 with comments.

2.6 - Encryption and EncryptionInfo

A key concept in the TDF specification is the ability to encrypt payloads, assertions, and keys. Whenever content is encrypted, encryption information must be provided. EncryptionInformation contains KeyAccess and EncryptionMethod information, providing the information necessary for decryption or key retrieval. Onion or layered encryption is also supported. In this case, there will be multiple EncryptionInformation elements within one EncryptionInformation group. Each EncryptionInformation has an optional sequenceNum attribute that is required to be provided when multiple EncryptionInformation elements are used. The order of sequence for encryption should be in increasing numerical order. The highest sequenceNum value corresponds to the outermost layer of encryption. For example, this layered or onion encryption may be required in a use case where both a system and a user must provide certificates before information can be decrypted. Encryption Method allows key size, algorithm, and Optimal Asymmetric Encryption Padding Scheme (OAEP)^[19] information.

2.7 - Linked or Embedded Data Objects

Linked objects classification does NOT impact the classification of the TDO. Embedded objects classification does impact the classification of the TDO.

2.8 - MIME type

The Multipurpose Internet Mail Extensions (MIME) type for a IC-TDF.XML document is application/dni-tdf+xml. This is a convention for our community. This type has NOT been registered with the Internet Assigned Numbers Authority (IANA). Should there be a conflict in the future it will be addressed at that time. Systems can use this MIME type to facilitate communications and address business needs within the community.

2.9 - 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.

2.10 - 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. IC-TDF.XML data validation constraint rule identifiers are prefixed with “IC-TDF-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 5](#). As the constraint rules are managed over time, IDs from deleted rules will not be reused.

Table 5 - 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 IC-TDF.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^[22] 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^[22] rules for this specification may be executed in *Oxygen*^[20] or with an XSLT 2.0-compliant processor using the XSLT 2.0^[28] 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^[27] and XSLT 2.0^[28] features. Regarding the use of XPath 2.0 and XSLT 2.0 with Schematron, the editor of the ISO Schematron standard stated the following:^[17]

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^[28] implementation of Schematron^[22] 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 IC-TDF.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^[25] 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 IC-TDF.XML model use Controlled Vocabulary Enumerations (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 6](#) 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 6 - Revision Constraints table

Validation Rules Version	11	12	13	13.201701	13.201804	201508	201609
Instance Version							
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 IC-TDF.XML schema can be found in a separate document inside the SchematronGuide directory, in the IC-TDF_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 IC-TDF.XML documents. The intent is to inform the development of systems capable of rendering or displaying IC-TDF.XML data for use by individuals not familiar with the details of the IC-TDF.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 IC-TDF.XML data rendering constraint rules.

Table 7 - 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 is considered conformant with the IC-TDF specification if it passes all of the following normative validation steps. The following steps do not dictate how this validation strategy is implemented.

4.1 - Definitions

Terms are defined the first time they are used. Definitions are cumulative, meaning that a term used in any given step may be defined in a previous step. The following definitions are global concepts, so they are defined in this section instead of in-line.

[Definition: A *TDF extension point* is an element within the IC-TDF specification whose purpose is to hold multiple forms of user content in-line.] There are six extension points within IC-TDF:

1. tdf:StringStatement
2. tdf:Base64BinaryStatement
3. tdf:StructuredStatement
4. tdf:StringPayload
5. tdf:Base64BinaryPayload
6. tdf:StructuredPayload

Note that tdf:ReferenceStatement and tdf:ReferenceValuePayload are not considered extension points because they only convey a link to content and do not hold content in-line.

[Definition: The content contained within elements tdf:Base64BinaryStatement and tdf:Base64BinaryPayload is referred to as *binary content*.]

[Definition: The content contained within elements tdf:StringStatement and tdf:StringPayload is referred to as *string content*.]

[Definition: The content contained within elements tdf:StructuredStatement and tdf:StructuredPayload is referred to as *structured content*.]

[Definition: The term *TDO structure* refers to all elements within an IC-TDF instance excluding the content of any TDF extension point].

4.2 - Why a verbose validation strategy is required

The IC-TDF specification is designed to be extremely flexible by allowing users to include several formats of in-line content in several extension points (see [Figure 8](#)). These *TDF extension points* require IC-TDF instances to use a more verbose validation strategy for several reasons:

1. IC-TDF schema defines the extension points tdf:StructuredStatement and tdf:StructuredPayload as <xs:any processContents="skip"/>, which skips all schema validation for the content contained within those extension points.

2. *Structured content* within the IC-TDF instance can contain data which can conflict with the data contained within the elements declared as part of the IC-TDF specification.

For example, the IC-TDF specification uses Information Security Markings (ISM) for conveying classification markings. The Publication Metadata (PUBS) specification also uses ISM. Suppose the payload contained an old PUBS document, which used a different version of ISM than defined in the IC-TDF specification. Applying the version of ISM business rules defined in IC-TDF to this instance document could easily fail because the older version ISM markings in the PUBS document could contain different attributes, removed tokens, among other changes.

3. For *binary content* and *string content*, XSD schema validation and XML business rules are not applicable and custom validation logic is required to validate that content.



Figure 8 : TDF Extension Points

4.3 - How to determine the ISM version within structured content

The version of ISM markings used within *structured content* is determined by the first occurrence of attribute @ism:DESVersion in document order contained in the structured content. If the

structured content does not specify attribute @ism:DESVersion, then the ISM version is defined to be the same as the ISM markings used within the parent IC-TDF structure (TDO or TDC).

4.4 - Required Order of Handling Assertions

Before any validation takes place on a TDO, a validation implementation **MUST** ensure that the TDO handling assertion is the first handling assertion in document order.

[Definition: The tdf:HandlingAssertion element which specifies attribute @tdf:scope with a value containing "TDC" is referred to as the *tdc handling assertion*.]

Before any validation takes place on a TDC, a validation implementation **MUST** ensure that the TDC handling assertion is the first handling assertion in document order.

[Definition: The ISM business rules define the first element in document order which specifies attribute @ism:resourceElement="true" to be the *resource element*.] The resource element contains the banner level ISM markings for the entire instance (i.e., the "roll-up").

The banner level markings within an IC-TDF instance are contained within a tdf:HandlingAssertion element and an instance may have multiple tdf:HandlingAssertion elements, each specifying a different scope. It is required that the first tdf:HandlingAssertion element in document order contain the banner level markings intended for the entire IC-TDF instance.

[Definition: The tdf:HandlingAssertion element which specifies attribute @tdf:scope with a value containing "PAYL" is referred to as the *payload handling assertion*]. [Definition: The tdf:HandlingAssertion element which specifies attribute @tdf:scope with a value containing "TDO" is referred to as the *tdo handling assertion*].

4.5 - TDO Validation Steps

This section outlines the required steps to fully validate a TrustedDataObject (TDO).

4.5.1 - Step 1 - TDO aware and cross assertion constraints

This step is intended to support validation which requires knowledge of the TDO structure.

IC-TDF validation, to include schema and business rules, should be run during this step.

ISM and NTK validation **MUST NOT** be run in this step because, as explained in the justification above, a *TDF extension point* could contain *structured content* which contains ISM or NTK markings from a different version of ISM/ NTK than the TDO structure is using, which could fail validation. ISM and NTK validation is performed in [Section 4.5.3 - Step 3 – TDO structure constraints](#). ARH and IC-EDH validation **SHOULD NOT** be performed at this step as it may be problematic when dealing with extension points that utilize different versions of these specifications from those used in the TDO.

TDO aware validation **MAY** be performed during this step. For example, one might want to run business rules specific to a certain domain or system. Some examples of custom validation could include:

- If this TDO contains an Assertion with child element X, then it must also contain a peer Assertion with child element Y.
- Verify that this TDO instance contains a custom assertion specific to a certain domain.
- Verify all bindings within this TDO.
- If the payload is encrypted, attempt to decrypt it and run additional custom validation on the decrypted content.

4.5.2 - Step 2 – Extension point constraints

This step is intended to support validation for the content of all *TDF extension points* contained within the TDO.

The child content of any *TDF extension point* MAY be validated. Any content validated in this step MUST be validated independently and in isolation. Determining which *TDF extension points* are validated in this step is implementation specific. For example, an implementation might choose to only validate *structured content* while ignoring *binary content* and *string content* completely. Or, an implementation might define a configuration which only validates *structured content* whose root element is in a certain namespace or set of namespaces.

If the content being validated is *structured content*, then the ISM business rules MUST NOT be applied unless the content is a *standalone ISM document*. [Definition: A *standalone ISM document* is an XML document which specifies the ISM attributes @ism:resourceElement and @ism:DESVersion]. Any NTK, ARH, or IC-EDH validation SHOULD be performed during this step for the *structured content* if the appropriate DESVersion attributes are specified.

Several examples of validation which could occur in this step include:

Schema and business rules for IC specifications from the 2012-Charlie release and earlier, including Publication Metadata (PUBS.XML) and Information Resource Metadata (IRM.XML).
Schema and business rules for mission specific assertion statements.
Custom validation for an audio/video file contained within a binary payload.

4.5.3 - Step 3 – TDO structure constraints

This step is intended to verify that ISM markings within the *TDO structure* are consistent. By treating *structured content* within *TDF extension points* as black boxes, only the ISM markings within the *TDO structure* will be validated. This includes ISM markings within HandlingAssertions and StatementMetadata. It does not include ISM markings within the payload and assertion extension points, which are considered 'black box' extensions in this step. This is also the time when any NTK, ARH, and IC-EDH validation that is specific to the *TDO structure* itself SHOULD be performed.

If IC-TDF rules were not run in Step 1:

[Definition: A *placeholder element* is an XML element whose localname is "PlaceHolderContent", namespace is "urn:placeholder", and contains no text content or child elements].

[Definition: A *TDF skeleton* is an IC-TDF instance in which the structured content contained within all TDF extension points has been replaced by a placeholder element]. Whether *string content* and *binary content* is preserved when converting an IC-TDF instance to a TDF skeleton is implementation specific. Replacing string content and binary content with default values may yield performance improvements during validation if that content is large in size and is not intended to be validated.

[Definition: A *TDF skeleton* whose root element is tdf:TrustedDataObject is referred to as a *TDO skeleton*].

The tdf:TrustedDataObject element MUST be converted into a *TDO skeleton*, which MUST be validated in isolation against the normative portions of the ISM specification version in use by the TDO. Additional validation MAY be performed during this step.

4.5.4 - Step 4 – ISM consistency constraints

This step is intended to verify that ISM markings contained within *structured content* matches the corresponding ISM markings within the *TDO structure*. This step has several sub-steps because assertions and payloads require slightly different processing depending upon certain criteria.

4.5.4.1 - Step 4a – Consistency constraints for Assertions with resource level portion markings

[Definition: An *assertion fragment* is a tdf:Assertion element containing at least one tdf:StatementMetadata element and a TDF extension point]. Whether an assertion fragment contains any other child elements (tdf:Binding, tdf:ReferenceList, etc) is implementation specific.

[Definition: A *structured assertion fragment* is an assertion fragment whose TDF extension point is tdf:StructuredStatement].

Structured assertion fragments meeting the following criteria MUST be validated in isolation against the normative portions of the ISM specification version in use by the TDO:

1. The *structured content* contains ISM markings.
2. The ISM markings contained in the *structured content* are from the same version of the ISM specification as the ISM markings within the *TDO structure*. See [Section 4.3 - How to determine the ISM version within structured content](#).
3. One of the tdf:StatementMetadata child elements specifies attribute @ism:resourceElement="true".

Validation of a *structured assertion fragment* verifies that the ISM markings contained within the *structured content* and the ISM markings contained within the tdf:StatementMetadata element are consistent. The ISM business rules use the tdf:StatementMetadata ISM markings as the resource level ("banner level") markings and treat the ISM markings in the *structured content* as portion markings. Constraint #3 above ensures that a tdf:StatementMetadata element can provide the resource level markings required for the ISM business rules.

For example, if the tdf:StatementMetadata contained @ism:classification="U" and the TDF extension point content contained @ism:classification="TS", then the ISM business rules would throw an error saying that unclassified documents must not contain TS portions.

4.5.4.2 - Step 4b – Consistency constraints for Payloads with resource level portion marking

[Definition: A *payload fragment* is a tdf:TrustedDataObject element containing a single tdf:HandlingAssertion element which is the payload handling assertion and a child TDF extension point]. Whether a payload fragment contains any other child elements (tdf:Assertion, etc) is implementation specific.

[Definition: A *structured payload fragment* is a payload fragment whose TDF extension point is tdf:StructuredPayload].

Structured payload fragments meeting the following criteria MUST be validated in isolation against the normative portions of the ISM specification version in use by the TDO.

1. The *structured content* contains ISM markings.
2. The ISM markings contained in the *structured content* are from the same version of the ISM specification as the ISM markings within the *TDO structure*. See [Section 4.3 - How to determine the ISM version within structured content](#).
3. The *payload handling assertion* specifies attribute @ism:resourceElement="true".

Validation of the structured payload fragment verifies that the ISM markings contained within the *structured content* are consistent with the ISM markings in the payload handling assertion. The ISM business rules use the *payload handling assertion* as the resource level ("banner level") markings and treats the ISM markings in the *structured content* as portion markings. Constraint #3 above ensures that the *payload handling assertion* can provide the resource level markings required for the ISM business rules.

For example, if the *payload handling assertion* contained @ism:classification="U" and the *structured content* contained @ism:classification="TS", then the ISM business rules would throw an error saying that unclassified documents must not contain TS portions.

4.5.4.3 - Step 4c – Consistency constraints for Assertions and Payloads with non-resource level markings

This step is intended to check the consistency of ISM markings within assertions and payloads which do not have corresponding resource level ISM portion markings in the TDO structure (assertions and payloads not checked in step 4a or 4b).

The tdf:TrustedDataObject element MUST be modified to replace *structured content* meeting the following criteria with a *placeholder element*.

1. The *structured content* contains ISM markings.

2. The ISM markings contained within the *structured content* are from a *different version* of the ISM specification as the ISM markings within the *TDO structure*. See [Section 4.3 - How to determine the ISM version within structured content](#).

The modified tdf:TrustedDataObject element MUST be validated in isolation against the normative portions of the ISM specification version in use by the TDO.

Replacing all of the *structured content* containing ISM markings from different versions allows the ISM business rules for the version used within the TDO structure to run correctly. The ISM business rules will use the tdo handling assertion as the resource level ("banner level") markings and treat the ISM markings in the rest of the TDO as portion markings. This step is very similar to [Section 4.5.3 - Step 3 - TDO structure constraints](#), but step 3 replaces all *structured content* with a *placeholder element* whereas this step leaves *structured content* in-line if it uses the same ISM version as the ISM markings within the TDO structure.

For example, if the *tdo handling assertion* contained @ism:classification="U" and the *structured content* of an assertion not checked in step 4a or 4b (using the same ISM version) contained @ism:classification="TS", then the ISM business rules would throw an error saying that unclassified documents must not contain TS portions.

4.6 - TDC Validation Steps

This section outlines the required steps to fully validate a TrustedDataCollection (TDC).

4.6.1 - Step 1 – TDC aware and cross assertion constraints

This step is intended to support validation which requires knowledge of the TDC structure.

IC-TDF validation to include schema and business rules should be run during this step.

ISM validation MUST NOT be run in this step because, as explained in the justification above, a *TDF extension point* could contain *structured content* which contains ISM markings from a different version of ISM than the TDC structure is using, which could fail validation. ISM validation is performed in [Section 4.6.3 - Step 3 - TDC structure constraints](#). NTK, ARH, and IC-EDH validation at this step may also be problematic when dealing with extension points that utilize versions of these specifications used in the TDO.

Additional validation may be performed during this step. For example, one might want to run business rules specific to a certain domain or system. Some examples of custom validation could include:

- Test if this TDC contains an Assertion with child element X, then it must also contain a peer Assertion with child element Y.
- Test if this TDC must contain a certain assertion type, such as a Multi-Audience Collection (MAC) assertion.

4.6.2 - Step 2 – Extension point constraints

This step is intended to support validation for the TDF extension point content contained within child tdf:Assertion elements of the TDC. The rules outlined in [Section 4.5.2 - Step 2 - Extension](#)

[point constraints](#) should be applied to each child tdf:Assertion element of the tdf:TrustedDataCollection element.

4.6.3 - Step 3 – TDC structure constraints

This step is intended to verify that ISM markings within the TDC structure are consistent. By treating *structured content* within *TDF extension points* as black boxes, only the ISM markings within the TDC structure will be validated. This includes ISM markings within HandlingAssertions and StatementMetadata. This is also the place to perform any NTK, ARH, and IC-EDH validation that is specific to the TDC structure itself.

[Definition: A TDF skeleton whose root element is tdf:TrustedDataCollection is referred to as a *TDC skeleton*].

The tdf:TrustedDataCollection element **MUST** be converted into a *TDC skeleton*, which **MUST** be validated in isolation against the normative portions of the ISM specification version in use by the TDC. Additional validation **MAY** be performed during this step.

4.6.4 - Step 4 – ISM consistency constraints

This step is intended to verify that ISM markings contained within *structured content* match the corresponding ISM markings within the TDC structure. This step has several sub-steps because assertions with resource level ("banner level") ISM markings require slightly different processing than non-resource level ISM markings.

4.6.4.1 - Step 4a – Consistency constraints for Assertions with resource level portion markings

This step is intended to verify the consistency of ISM markings contained within child tdf:Assertion elements of the tdf:TrustedDataCollection element. The rules outlined in [Section 4.5.4.1 - Step 4a – Consistency constraints for Assertions with resource level portion markings](#) should be applied to each child tdf:Assertion element within the TDC.

4.6.4.2 - Step 4b – Consistency constraints for Assertions with non-resource level markings

This step is intended to check the consistency of ISM markings within child tdf:Assertion elements which do not have corresponding resource level ISM portion markings in the TDC structure (assertions not checked in step 4a).

The tdf:TrustedDataCollection element **MUST** be modified to replace *structured content* meeting the following criteria with a *placeholder element*:

1. The *structured content* contains ISM markings.
2. The ISM markings contained within the *structured content* are from a different version of the ISM specification as the ISM markings within the TDC structure. See [Section 4.3 - How to determine the ISM version within structured content](#).

The modified tdf:TrustedDataCollection element MUST be validated in isolation against the normative portions of the ISM specification version in use by the TDC.

Replacing all of the *structured content* containing ISM markings from different versions allows the ISM business rules for the version used within the TDC structure to run correctly. The ISM business rules will use the tdc handling assertion as the resource level (“banner level”) markings and treat the ISM markings in the rest of the TDC as portion markings.

For example, if the TDC handling assertion contained @ism:classification=”U” and the structured content of an assertion not checked in step 4a (using the same ISM version) contained @ism:classification=”TS”, then the ISM business rules would throw an error saying that unclassified documents must not contain TS portions.

4.6.5 - Step 5 - Recursive Validation

A tdf:TrustedDataCollection element supports recursion by allowing child tdf:TrustedDataObject and tdf:TrustedDataCollection elements. Each tdf:TrustedDataObject element must be validated according to the steps outlined in [Section 4.5 - TDO Validation Steps](#). Each tdf:TrustedDataCollection element must be validated according to the steps outlined in [Section 4.6 - TDC Validation Steps](#).

Chapter 5 - Generated Guides

5.1 - Schema Guide

The detailed description and reference documentation for the IC-TDF.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 IC-TDF.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 IC-TDF.XML Schematron rules can be found in a separate document named *IC-TDF_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.

Chapter 6 - Future Features

6.1 - Explicit Scope

In future versions, the concept of scope will be extended to support a flexible, explicit list of elements. The token [EXPLICIT] is expected to be used to indicate this granularity. An assertion using explicit scope will require either a ReferenceList or a BoundValueList and the elements to which it "applies" will be determined by the values in the ReferenceList or BoundValueList.

6.2 - BoundValueList

A key concept in the TDF specification is the ability to cryptographically assure the relationship among portions of the document. Future versions of TDF will make Cryptographic Binding more flexible and granular through the introduction of an optional Bound Value List as a child of the **Binding** element. A **BoundValueList** is a container of bound value references that point to the elements that are included in a cryptographic binding. The **idref** attribute of **BoundValue** or **Reference** element is the internal instance reference to the element being bound. The intent of the **BoundValueList** is to allow granular control over the scope of the binding signature. In the future, when BoundValueList is present, the **SignatureValue** will be calculated over the normalized value of the **BoundValueList** using the normalization method denoted in the **Binding/SignatureValue/@normalizationMethod** attribute.

In IC-TDF, where the **BoundValueList** is not present, the **SignatureValue** is always calculated over a concatenation of the normalized portions of the document in the same order they appear in the document described by the Assertion.

The normalization method expressed in **Binding/SignatureValue/@normalizationMethod** and **Binding/BoundValueList/BoundValue/@normalizationMethod** is a URI that provides guidance on how to format the included values such as whitespace, attributes, and child nodes in a universally consistent manner. The normalization method is essential to prevent formatting such as white-space and order from interfering with the validation of the cryptographic integrity of data. For example, XML canonicalization is one form of normalization that might be utilized. More information on XML canonicalization is available online at: [W3C Canonical XML](http://www.w3.org/TR/xml-c14n) [http://www.w3.org/TR/xml-c14n]. To use XML canonicalization as a normalization method, provide the URI to the form of XML canonicalization you are using, such as <http://www.w3.org/TR/2001/REC-xml-c14n-20010315> [http://www.w3.org/TR/2001/REC-xml-c14n-20010315] as the value for the **Binding/SignatureValue/@normalizationMethod**. This example URL is the URL defined in XML-SEC Rec for inclusive c14n without comments.

Appendix A Feature Summary

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

Table 8 - TDF Dependency over Time

Dependent DES	V1	V2	V3	V2014-DEC	V2014-DECr2017-JUL
ISM*	V9	V9+	V9+	V9+	V9+
IC-EDH*	V1	V1+	V1+	V1+	V1+
NTK*	V7	V7+	V7+	V7+	V7+
ARH*	V1	V1+	V1+	V1+	V1+
RevRecall*	N/A	N/A	N/A	V2014-DEC+	V2014-DEC+
USAgency					V2016-SEP+
MN					V2015-AUG+
LIC					V2015-AUG+
ISMCAT					V2017-JUL+
IC-ID					V1+

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

Table 9 - 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. IC-TDF Feature Summary

Table 10 - IC-TDF Feature comparison

IC-TDF Feature Comparison						
Required date	Feature	V1	V2	V3	2014-DEC	2014-DECr2017-JUL
	Mime Types	F	F	F	F	F
	Multiple versions of ISM.XML (V9 - Current)	N	F	F	F	F
	Multiple versions of NTK.XML (V7 - Current)	N	F	F	F	F
	Multiple versions of ARH.XML (V1 - Current)	N	F	F	F	F
	Multiple versions of IC-EDH.XML (V1 - Current)	N	F	F	F	F

IC-TDF Feature Comparison						
Required date	Feature	V1	V2	V3	2014-DEC	2014-DECr2017-JUL
	TDC scope [PAYL]	F	N/A	N/A	N/A	N/A
	TDC scopes [DESC_TDO], [DESC_PAYL], and [TDC_MEMBER]	N	F	F	F	F
	Multiple bindings in Assertions and HandlingAssertions	N	F	F	F	F
	Version decoupling, allowing import of any version of ISM and other dependent specifications at or above ISM v9+, NTK v7+, ARH v1+, and IC-EDH v1+	N	F	F	F	F
	Vector encryption	N	N	N	F	F

Appendix B Change History

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

Table 11 - DES Version Identifier History

Version	Date	Purpose
1	17 July 2012	Initial Release
2	21 January 2013	Routine revision to technical specification. For details of changes, see Section B.4 - V2 Change Summary
3	16 August 2013	Routine revision to technical specification. For details of changes, see Section B.3 - V3 Change Summary
2014-DEC	4 December 2014	Routine revision to technical specification. For details of changes, see Section B.2 - V2014-DEC Change Summary
2014-DEC-r2017-JUL	21 July 2017	Routine revision to technical specification. For details of changes, see Section B.1 - V2014-DEC-r2017-JUL Change Summary

B.1 - V2014-DEC-r2017-JUL Change Summary

Significant drivers for Version 2014-DEC-r2017-JUL include:

- Community Change Requests

The following table summarizes the changes made to 2014-DEC in developing V2014-DEC-r2017-JUL.

Table 12 - Data Encoding Specification V2014-DEC-r2017-JUL Change Summary

#	Change	Artifacts changed	Compatibility Notes
1	Change the IC-TDF-ID-00046 Rule to Handle ism:DESVersion Values with dash "-" token separators.(CR-2016-081)	Schematron CompareVersionsInSkel etion revised	Allows DESVersion attribute to contain version and revision numbers separated by a "-".

#	Change	Artifacts changed	Compatibility Notes
2	Reorganize IC-TDF Schematron Rules Folder To Handle Deleted Rules.(CR-2016-084)	Schematron IC-TDF-ID-00020 deleted IC-TDF-ID-00021 deleted IC-TDF-ID-00022 deleted IC-TDF-ID-00023 deleted IC-TDF-ID-00024 deleted	Simplifies processing of unit tests for schematron rules.
3	Changed "TDO" to "TDC" in rule text of IC-TDF-ID-00005. (CR-2017-025)	Schematron IC-TDF-ID-00005 modified	Minimal impact to generation and ingestion systems.
4	Referenced the "Assertion Scopes" section in Chapter 2 of the IC-TDF DES document in the scope reference documentation. (CR-2016-008)	Documentation Schema	No impact to generation and ingestion systems.
5	Bug in TDF Rule 00014 in v2014v12; allow tdh:EncryptionInformation elements to be nested in tdf:WrappedKey elements. (CR-2017-016)	Schematron IC-TDF-ID-00014 modified	Data generation and ingestion systems need to be updated to use the modified schematron rules.
6	There were a few typos throughout the documents. The mistakes were things like "guarenteed", "encyrption", "pertinate", "identifer" and "encapslating" and others. They were all in the comment sections. (CR-2017-099)	Documentation Schema	No impact to generation and ingestion systems.
7	Added IC-TDF-ID-00055 Rule to enforce at most 1 handling assertion scoped PAYL containing EDH for unencrypted TDO (CR-2016-037)	Schematron IC-TDF-ID-00055 added	No impact to generation and ingestion systems.
8	Create JSON version of CVEs in IC-TDF (CR-2017-054)	CVEs	No impact to systems.

#	Change	Artifacts changed	Compatibility Notes
9	Create CSV version of CVEs in IC-TDF (CR-2017-032)	CVEs	No impact to systems.
10	Updated tdf:version enforcement rule to be warning and handle trailing version text (CR-2017-082, CR-2017-027)	Schema Schematron IC-TDF-ID-00054 added IC-TDF_XML.sch modified	Data generation and ingestion systems need to be updated to accommodate the changes to the rules.
11	Added inverse dependency section and definitions for Dependencies and Inverse Dependencies. (CR-2017-112)	Documentation	No impact to systems.
12	The schema change logs will no longer be maintained as of the 2017-JUL release. The existing change logs will only serve as legacy information. For changes to schema as of and after 2017-JUL, reference the change history in the DES.	Schema	No impact to systems.
13	Added the revision constraint section since this is the first revision of ISM.	Documentation	Data generation and ingestion systems will may need to be updated to properly validate against the right revisions of specifications.
14	Updated rule IC-TDF-ID-00042 to require that the first handling assertion include an EDH. (CR-2017-142)	Schematron IC-TDF-ID-00042 modified	Data generation and ingestion systems will may need to be updated to properly position a RevisionRecall assertion.
15	Updated rule IC-TDF_ID_00017 to properly require an EDH with Scope TDC to have @ism:resourceElement="true". This was aligning IC-TDF_ID_00017 with the existing logic in IC-TDF_ID_00016 which had done it correctly for TDOs. (CR-2017-198)	Schematron IC-TDF-ID-00017 modified	Data generation and ingestion systems will may need to be updated to properly ensure the first assertion has @ism:resourceElement="true"
16	Enable use of ARH or EDH instead of only EDH for describing the classification of Encrypted assertions. (CR-2017-202)	Schematron IC-TDF-ID-00030 modified	Data generation and ingestion systems will may need to be updated to properly allow and process ARH for security of encrypted assertions.

#	Change	Artifacts changed	Compatibility Notes
17	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.

#	Change	Artifacts changed	Compatibility Notes
18	Updated rule documentation to remove use of “we”. (CR-2017-208)	Schematron IC-TDF-ID-00001 modified IC-TDF-ID-00002 modified IC-TDF-ID-00003 modified IC-TDF-ID-00004 modified IC-TDF-ID-00005 modified IC-TDF-ID-00006 modified IC-TDF-ID-00007 modified IC-TDF-ID-00008 modified IC-TDF-ID-00009 modified IC-TDF-ID-00010 modified IC-TDF-ID-00011 modified IC-TDF-ID-00012 modified IC-TDF-ID-00013 modified IC-TDF-ID-00014 modified IC-TDF-ID-00015 modified IC-TDF-ID-00017 modified	No impact to systems.

#	Change	Artifacts changed	Compatibility Notes
		IC-TDF-ID-00018 modified IC-TDF-ID-00019 modified IC-TDF-ID-00025 modified IC-TDF-ID-00026 modified IC-TDF-ID-00027 modified IC-TDF-ID-00032 modified IC-TDF-ID-00036 modified IC-TDF-ID-00037 modified IC-TDF-ID-00039 modified IC-TDF-ID-00041 modified IC-TDF-ID-00045 modified IC-TDF-ID-00055 modified	
19	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.
20	Update the version numbering EBNF to reflect the existence of Revisions. (CR-2017-237)	Documentation	No impact to systems.

B.2 - V2014-DEC Change Summary

Significant drivers for Version 2014-DEC include:

- Addition encryption algorithm support

The following table summarizes the changes made to V3 in developing V2014-DEC.

Table 13 - Data Encoding Specification V2014-DEC Change Summary

Change	Artifacts changed	Compatibility Notes
Changed DESVersion to represent the year and month of release. Also allowed for extension of specification by adding a '-' followed by a string to denote a custom implementation.	DES Schema Schematron IC-TDF-ID-00036 revised IC-TDF-ID-00037 revised IC-TDF-ID-00045 revised	Data generation and ingestion systems need to be updated to use the modified version numbering and schematron rules.
Improved encryption support by allowing for Initialization Vector, tweak, nonce, hash algorithm, Mask Generation Function, Additional Authentication Data, authentication tag, key encoding format, and Provable Data Possession wrapped keys.	Schema	Data generation and ingestion systems need to be updated to support the new components.
Required HandlingAssertions to come first in a TDO/TDC and required that the HandlingAssertion scoped TDO come first in a TDO.	Schema Schematron IC-TDF-ID-00042 Added	Data generation and ingestion systems need to be updated enforce the proper ordering.
Added rule to enforce presence of NTK at "top" level if NTK is present in any part of the TDF skeleton structure.	Schematron IC-TDF-ID-00043 Added IC-TDF-ID-00044 Added	Data generation and ingestion systems may need to be updated correctly place NTK.
Added Version Declarations section to describe handling and inheritance of specification versions in assertions and payloads.	Documentation	Data generation, ingestion, or manipulation systems may need to be updated to properly handle version declarations.

Change	Artifacts changed	Compatibility Notes
Updated Schema and Schematron rules to deal with Revision Recall handling assertion.	Schematron Schema IC-TDF_ID_00004 Changed IC-TDF_ID_00005 Changed IC-TDF_ID_00016 Changed IC-TDF_ID_00045 Added	Data generation and ingestion systems will need to be updated to use the new Revision Recall handling assertion.

B.3 - V3 Change Summary

Significant drivers for Version 3 include:

- Improve support of Onion encryption
- Support of Suite-B encryption

The following table summarizes the changes made to V2 in developing V3.

Table 14 - Data Encoding Specification V3 Change Summary

Change	Artifacts changed	Compatibility Notes
Updated the EncryptionInformation group to support onion encryption and Suite-B algorithms.	Documentation Schema Schematron IC-TDF-ID-00040 Added IC-TDF-ID-00041 Added	Data generation and ingestion systems need to be updated to understand the new schema structure.

Change	Artifacts changed	Compatibility Notes
Removed the value 'TDO PAYL' as an allowable value from the enumeration for the scope attribute. Removed the schematron rules that were looking for the 'TDO PAYL' scope.	Schema Schematron IC-TDF-ID-00020 Removed IC-TDF-ID-00021 Removed IC-TDF-ID-00022 Removed IC-TDF-ID-00023 Removed IC-TDF-ID-00024 Removed	Data generation and ingestion systems need to be updated to the new schema structure and to no longer enforce the schematron rules.

B.4 - V2 Change Summary

Significant drivers for Version 2 include:

- See ISM V10 drivers
- See EDH V2 drivers

The following table summarizes the changes made to V1 in developing V2.

Table 15 - Data Encoding Specification V2 Change Summary

Change	Artifacts changed	Compatibility Notes
Added Schematron rules to require the specification of the issuer attribute and either the subject or serial attribute for the tdf:Signer element.	Schematron IC-TDF_ID_00038.sch	Data generation and ingestion systems need to be updated enforce the new rules.
Added Schematron rules to ensure that the versions of the imported specs meet the minimum allowed versions.	Schematron IC-TDF-ID-00036 Added IC-TDF-ID-00037 Added	Data generation and ingestion systems need to be updated enforce the new rules.

Change	Artifacts changed	Compatibility Notes
Updated the GUIDE id in the example files to comply with the updated regex in IC-EDH-ID-00007. The updated rule ensures there are no additional characters before or after the id.	Examples	Data generation and ingest systems complying with the GUIDE id rules do not need to be updated. Systems that were allowing invalid GUIDE ids will need to be updated to comply with the constraint rule.
Added validation strategy to the DES Version.	DES	Systems performing validation of the TDF should follow the appropriate validation strategy to ensure thorough and complete validation.
Added requirements for References to have external security markings.	IC-TDF-ID-00033 added IC-TDF-ID-00034 added	Data generation and ingest systems will be required to comply with the new rules.
Added scopes [DESC_TDO], [DESC_PAYL], and [TDC_MEMBER] for use within TDC Assertions to disambiguate trusted data collection scope meaning.	Schema IC-TDF-ID-00007 modified IC-TDF-ID-00035 added	Data generation and ingest systems will be required to comply with the new rules.
Deprecated scope [PAYL] for use within TDC Assertions.	IC-TDF-ID-00007 modified	Data generation and ingest systems will be required to comply with the new rules.
Added support for multiple bindings within Assertions and HandlingAssertions.	Schema DES	Data generation and ingest systems need to be updated to support the new schema structure.
Version decoupling, allowing import of any version of ISM and other dependent specifications at or above ISM v9+, NTK v7+, ARH v1+, and IC-EDH v1+.	DES	Data ingestion systems need to be aware of this change and ensure they check appropriate dependent spec versions for validation.
Updated Schema to ISM v10.	Schema	Updated the Schema itself to use ism:DESVersion to 10 to mark the xsd schema instance with classification markings.
Added rule to only allow HandlingAssertions with scope of payload to use of the appliesToState attribute because only the payload can have encrypted or unencrypted states.	Schematron IC-TDF-ID-00039 added	Data generation and ingest systems will be required to comply with the new rules, however this rule should prevent systems from having to deal with a nonsensical case.

Appendix C List of Abbreviations

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

ADD	Abstract Data Definition
ARH	Access Rights and Handling
CVE	Controlled Vocabulary Enumeration
DES	Data Encoding Specification
DNI	Director of National Intelligence
EDH	Enterprise Data Header
ESB	Enterprise Standards Baseline
FOUO	For Official Use Only
HTML	HyperText Markup Language
IANA	Internet Assigned Numbers Authority
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
IC-ID	IC Identifier
ICD	Intelligence Community Directive
ICPM	Intelligence Community Policy Memorandum
ICS	Intelligence Community Standard
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IRM	Information Resource Metadata
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
LIC	License
MAC	Multi Audience Collection
MIME	Multipurpose Internet Mail Extensions
MN	Mission Need Profile
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
TDC	Trusted Data Collection
TDF	Trusted Data Format
TDO	Trusted Data Object
USAGENCY	Controlled Vocabulary Enumeration Encoding Specification for US Agencies
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
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] ARH.XML

Office of the Director of National Intelligence. *XML Data Encoding Specification for Access Rights and Handling (ARH.XML)*. Available online Intelink-TS at: <http://go.ic.gov/Fub6Gnw>
Available online Intelink-U at: <https://w3id.org/ic/standards/ARH>
Available online at: <https://w3id.org/ic/standards/public>

[4] 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>

[5] IC-EDH.XML

Office of the Director of National Intelligence. *XML Data Encoding Specification for Enterprise Data Header (IC-EDH.XML)*. Available online Intelink-TS at: <http://go.ic.gov/TQjVx3d>
Available online Intelink-U at: <https://w3id.org/ic/standards/EDH>
Available online at: <https://w3id.org/ic/standards/public>

[6] 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>

[7] 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

[8] 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>

[9] 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

[10] ICD 501

Office of the Director of National Intelligence. *Discovery and Dissemination or Retrieval of Information within the Intelligence Community*. Intelligence Community Directive 501. 21 January 2009.

Available online Intelink-TS at: <http://go.ic.gov/GG61roi>

Available online at: http://www.dni.gov/files/documents/ICD/ICD_501.pdf

[11] 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>

[12] ICPM 2007-200-2

Office of the Director of National Intelligence. *Preparing Intelligence to Meet the Intelligence Community's Responsibility to Provide*. Intelligence Community Policy Memorandum 2007-200-2. 11 December 2007.

Available online at: <http://www.dni.gov/files/documents/IC%20Policy%20Memos/ICPM%202007-200-2%20Responsibility%20to%20Provide.pdf>

[13] 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>

[14] 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>

[15] 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>

[16] 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>

[17] Jelliffe

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

[18] NTK.XML

Office of the Director of National Intelligence. *XML Data Encoding Specification for Need-To-Know Metadata (NTK.XML)*.
Available online Intelink-TS at: <http://go.ic.gov/YLXsYUX>
Available online Intelink-U at: <https://w3id.org/ic/standards/NTK>
Available online at: <https://w3id.org/ic/standards/public>

[19] OAEP

Mihir Bellare, Phillip Rogaway. *Optimal Asymmetric Encryption Padding Scheme (OAEP)*.
Available for purchase at: <http://dx.doi.org/10.1007/BFb0053428>
Conference online at: <http://www.informatik.uni-trier.de/~ley/db/conf/eurocrypt/eurocrypt94.html>

[20] Oxygen

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

[21] REVRECALL.XML

Office of the Director of National Intelligence. *XML Data Encoding Specification for Revision Recall (RevRecall.XML)*.
Available online Intelink-TS at: <http://go.ic.gov/8HB0HUh>
Available online Intelink-U at: <https://w3id.org/ic/standards/REVRECALL>
Available online at: <https://w3id.org/ic/standards/public>

[22] 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/>

[23] 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>

[24] 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>

[25] 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>

[26] 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>

[27] 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/>

[28] 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.^[13]