



Intelligence Community Technical Specification

Intelligence Community Specification Framework

Version 2021-NOV

December 1, 2022

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 / Scope / History	1
1.2 - Authority	1
1.3 - Background	1
1.4 - Audience and Applicability	2
Chapter 2 - Development Guidance	3
2.1 - Schema Fragments	3
2.2 - Data Validation Constraint Rules	3
2.2.1 - Purpose	3
2.2.2 - Additional Constraints	3
2.2.2.1 - DES Constraints	3
2.2.3 - Constraint Rules	3
2.3 - Trusted Data Format	3
2.3.1 - Packaging Dependency versus Real Dependency	4
2.3.2 - SOME-TDF Reference in Related Specification Diagrams	4
2.3.3 - Environment Validation Schematron	4
Chapter 3 - Governance and Management	5
3.1 - IC CIO Authorities	5
3.2 - CMSTT and ESTT Working Groups	5
3.3 - Enterprise Standards Baseline (ESB) Management	6
3.4 - CMSTT and ESTT Specification Management	9
Chapter 4 - Specification Conventions	11
4.1 - Language	11
4.2 - Terminology	11
4.3 - Typography	11
4.4 - Dependency Definitions	11
4.4.1 - IC-SF Dependencies	12
4.4.2 - Dependency Diagrams	12
4.5 - Package Types	12
4.5.1 - Standalone Packages	12
4.5.2 - Convenience Packages	12
4.5.3 - Convenience Light Packages	13
4.6 - Conformance	13
4.7 - Version Policies	14
4.7.1 - XML Namespace Policy	14
4.7.2 - Version Numbering	14
4.7.2.1 - Trusted Data Format Versions	16
Chapter 5 - Specification Overview	17
5.1 - Specification Architecture Approach	17
5.1.1 - Specification Locations	18
5.2 - Components of Access Control Decisions	20
5.3 - Type of Specifications	21
5.3.1 - Data Encoding Specifications (DES)	21
5.3.2 - CVE Encoding Specifications (CES)	22
5.3.3 - Taxonomy Encoding Specifications (TES)	22
5.3.4 - Abstract Data Definitions (ADD)	22

5.3.5 - Access Control Encoding Specifications (ACES)	22
5.3.6 - Attribute Practice Compliance Statement (APCS)	22
5.4 - Components of Specifications	23
5.4.1 - Main Documents and Annexes	23
5.4.2 - Signature Memo	23
5.4.3 - Controlled Value Enumerations (CVE)	23
5.4.3.1 - XML Notes	23
5.4.3.2 - CSV Notes	23
5.4.3.3 - JSON Notes	24
5.4.3.4 - RELAX NG Notes	24
5.4.4 - Examples	24
5.4.5 - Schema	25
5.4.6 - SchemaGuide	25
5.4.7 - Constraint Rules (Schematron)	26
5.4.8 - Taxonomies	26
5.4.9 - XSL	26
5.4.10 - manifest.md5	27
5.4.11 - README.xhtml	27
5.4.12 - XXXpackage.properties	27
5.5 - Families of Specifications	27
5.5.1 - Attribute Based Access Control (ABAC) Metadata and Guidance	27
5.5.2 - Textual Documents	27
5.5.3 - Trustable Exchange & Micro Assertions	27
5.5.4 - Descriptive Metadata	28
5.5.5 - Enterprise Audit	28
5.5.6 - Guidance/Infrastructure	28
5.5.7 - Controlled Vocabulary Enumerations	28
5.5.8 - Abstract Data Definitions (ADDs)	28
5.5.9 - Service Security	28
Chapter 6 - Constraints	29
6.1 - Types of Constraint Rules	29
6.1.1 - Validation Constraint Rules	29
6.1.2 - Rendering Constraint Rules	29
6.1.3 - "Living" Constraint Rules	29
6.1.4 - Classified or Controlled Constraint Rules	30
6.2 - Constraint Terminology	30
6.3 - Errors and Warnings	30
6.4 - Rule Identifiers	30
6.5 - Data Validation Constraint Rules	31
6.5.1 - Purpose	31
6.5.2 - Schematron	31
6.5.3 - Non-null Constraints	32
6.5.4 - Value Enumeration Constraints	32
6.5.5 - Additional Constraints	32
6.5.5.1 - Version Constraints	32
6.5.5.2 - Revision Constraints	32
6.6 - Data Rendering Constraint Rules	34
6.7 - Conformance Validation	34
6.7.1 - Schema Validation	34

6.7.2 - Business Rule Validation	35
Chapter 7 - Generated Guides	36
7.1 - Schema Guide	36
7.2 - Schematron Guide	37
Appendix A - Change History	38
A.1 - V2021-NOV Change Summary	38
A.2 - V2021-JAN Change Summary	38
A.3 - V2019-MAR Initial Release Summary	39
Appendix B - Glossary	40
Appendix C - List of Abbreviations	41
Appendix D - Bibliography	44
Appendix E - Points of Contact	50
Appendix F - IC CIO Approval Memo	51

List of Figures

Figure 1 - Working Group Relationships	6
Figure 2 - Three-legged Stool of Access Decisions	20

List of Tables

Table 1 - Numerical Rule Identifier Ranges	30
Table 2 - Revision Constraints table	34
Table 3 - DES Version Identifier History	38
Table 4 - Data Encoding Specification 2021-NOV Change Summary	38
Table 5 - Data Encoding Specification V2021-JAN Change Summary	39
Table 6 - Data Encoding Specification V2019-MAR Initial Release Summary	39

Chapter 1 - Introduction

1.1 - Purpose / Scope / History

The *Intelligence Community Specification Framework* (IC-SF.XML) defines the basic conceptual structure and outlines the core philosophy of Intelligence Community (IC) technical specifications. This framework is applicable to the IC and information produced by, stored, or shared within the IC. This framework may have relevance outside the scope of intelligence; however, prior to applying outside of this defined scope, the framework should be closely scrutinized and differences separately documented and assessed for applicability.

Over the years, the number of IC technical specifications has grown substantially for various reasons, from addressing new enterprise needs to isolating independent concepts into separate specifications. Within these specifications, there are many aspects that are universal to all of the IC specifications, or at least applicable to a particular family or format of specification. In a desire to simplify the specifications and make them easier to understand by those who are engineering and implementing enterprise exchange systems, this framework was born.

The intent of this specification is to provide the foundational information in one centralized location. It was identified that, by cluttering each specification with the core foundational information, unique and important components of the specifications were overlooked. In addition, this framework contains an XML Schema Definition (XSD) schema, containing fragments used in multiple specifications. More information on these fragments may be found in [Section 2.1 - Schema Fragments](#). This framework also goes beyond the foundational components of the specifications by addressing the processes by which the specifications are managed and published. It also details, with more depth, how the specifications interact with one another based on the types of specifications defined in [Section 5.3 - Type of Specifications](#).

1.2 - Authority

Intelligence Community Standard (ICS) 500-20, *Intelligence Community Enterprise Standards Compliance* ^[16], defines the Intelligence Community Enterprise Standards Baseline (IC ESB) and the applicability of such to an IC element. Department of Defense (DoD) Instruction 8310.01, *Information Technology Standards in the DoD* ^[6] and DoD Instruction 8320.02, *Data Sharing in a Net-Centric Department of Defense* ^[7], require DoD elements to use the, *DoD IT Standards Registry* (DISR^[5]).

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* ^[15] 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 ICS 500-21, *Tagging of Intelligence and Intelligence-Related Information* [\[17\]](#) 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* (ADD.XML [\[2\]](#)). Many IC encoding specifications are based on XML, but other technologies are possible. For example, *Text and XML Data Encoding Specification for Intelligence Community Identifier* (IC-ID.XML [\[11\]](#)) defines a plain-text format for IC Identifiers as well as an associated XML structure.

1.4 - Audience and Applicability

This framework is intended for those that need to use and understand the basic structure of and philosophy behind intelligence community technical specifications.

Chapter 2 - Development Guidance

2.1 - Schema Fragments

IC-SF.XML contains XSD schema fragments that may be imported into other specifications. Currently, specifications that need hash verification will import the schema fragments from IC-SF.XML in order to verify contents of a payload.

2.2 - Data Validation Constraint Rules

2.2.1 - Purpose

The IC-SF.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. For more information, please see [Section 6.5 - Data Validation Constraint Rules](#).

2.2.2 - Additional Constraints

2.2.2.1 - DES Constraints

The Data Encoding Specification (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, Controlled Vocabulary Enumeration (CVE)s and business rules are intended by the author to be used.

2.2.3 - Constraint Rules

The detailed constraint rules for the IC-SF.XML schema can be found in a separate document inside the Documents/IC-SF directory, in the "IC-SF_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 "IC-SF_Rules.pdf" file.

2.3 - Trusted Data Format

Prior to the 2021-JAN release, *XML Data Encoding Specification for Trusted Data Format IC-TDF.XML*^[13] was the only specification that handled Trusted Data Format (TDF) profiles. *XML Data Encoding Specification for Trusted Data Format - Base* (BASE-TDF.XML^[4]), introduced during the 2021-JAN release, is the foundational specification from which all other TDF specifications are derived. TDF specifications, such as IC-TDF.XML^[13], inherit only those capabilities that are needed to fulfill its requirements.

2.3.1 - Packaging Dependency versus Real Dependency

There are many IC specifications that depend on some TDF specification since they exist as assertions inside a Trusted Data Object (TDO). This kind of dependency is a packaging dependency where some TDF specification is needed so that there is an example of how a specific assertion specification is used. This is different than a real dependency where a specification is dependent on another specification in its schema or schematron.

2.3.2 - SOME-TDF Reference in Related Specification Diagrams

Because there are multiple TDF specifications that an assertion specification can use, instead of identifying a specific TDF specification in the assertion specification's related specification diagram, a generic SOME-TDF reference will be used. SOME-TDF is not an actual specification, just a placeholder in the diagram to show that an assertion specification will depend on some TDF specification in its use.

2.3.3 - Environment Validation Schematron

Because there are multiple TDF specifications that an assertion specification can use, there can no longer be environment validation schematron for a specific TDF specification since that would cause validation errors when a different TDF specification is used.

Chapter 3 - Governance and Management

3.1 - IC CIO Authorities

The governance of an IC specification and the structure and philosophy it describes, including any requirement to use the specification or prohibition thereof, is explicitly outside the scope of the specification. ICS 500-20^[16], defines the IC ESB and the applicability of such to an IC element. DoD Instruction 8310.01^[6] and DoD Instruction 8320.02^[7], require DoD elements to use the DoD Information Technology Standards Registry (DISR) tool^[5] and process defined in [Section 3.3 - Enterprise Standards Baseline \(ESB\) Management](#).

Use of an IC specification MUST be consistent with applicable Federal statutes, Executive Orders, Presidential Directives, Attorney General approved guidelines, IC Policy, IC element policies, established concepts of operation, agreements, contractual obligations, etc. However, the determination of any such requirements or restrictions is the sole responsibility of each implementing entity. Implementers may wish to consult the Office of General Counsel for their cognizant agency to determine existing requirements and restrictions for the use of a specification and to determine if new agreements or policy changes are required related to the use of a specification.

3.2 - CMSTT and ESTT Working Groups

In order to keep the standards relevant and useful, topic-focused working groups, forums, and tiger teams have been established. These are created and disbanded as necessary and are tasked with selecting or developing the technical specifications, to include collaborating on writing, reviewing, and testing. These working groups also serve as communication mechanisms to ensure that IC and DoD elements are provided the information they need to concur with and implement community standards in a way that facilitates information exchange. The Data Standards Coordination Activity (DSCA) is a forum for collaboration and coordination of data and metadata standards, specifications, and profiles of common concern produced by the IC CIO. Participants come from all IC and many DoD elements and require a Government sponsor. [Figure 1](#) depicts the relationship between the DSCA and its working groups.

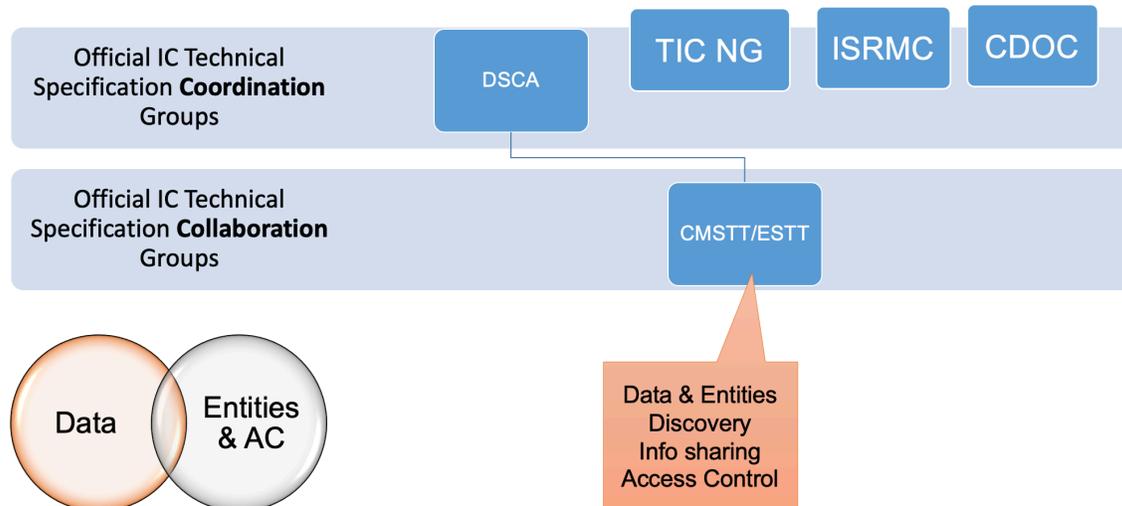


Figure 1 : Working Group Relationships

- *Data Standards Coordination Activity (DSCA)*: Facilitates the establishment and promulgation of common information sharing data standards. The DSCA gains its authority from the IC CIO. Under the Joint Enterprise Standards Committee (JESC), it functions as the Data Technical Working Group (TWG).
- *Common Metadata Standards Tiger Team (CMSTT)* : Facilitates community discussion on various data-related specifications produced by the IC CIO. Meetings are held monthly and are combined with the ESTT.
- *Entity Standards Tiger Team (ESTT)* : Facilitates community discussion on various entity-related specifications produced by the IC CIO. Entity specifications define identity, authentication and access control attributes for person and non-person entities. Meetings are held monthly and are combined with the CMSTT.

3.3 - Enterprise Standards Baseline (ESB) Management

The IC ESB serves as the collection of enterprise standards against which implementations of the IC enterprise architecture will be assessed to determine compliance. The DISR is the single, unifying DoD registry for approved IT and National Security System (NSS) standards and standards profiles. All entries into the DISR and IC ESB baselines will be entered through the DISR^[5] tool.

The JESC is a GS-15/O-6-level activity for validating and adjudicating IT standards and admitting new standards into the DISR and IC ESB. The JESC includes involvement from both the DoD and IC. The JESC advises, supports, and submits recommendations to the appropriate DoD and IC senior governance authorities. It standardizes procedures for categorizing, documenting, adopting, and implementing enterprise standards, profiles, and specifications while eliminating duplicative aspects of the respective DoD and IC enterprise standards baselines. To do this, the DoD and the IC have been evolving the standards lifecycle management. In particular, there are twelve TWGs, broken into two categories, Core and Domain, that manage the IT standards and must meet one or more of the following criteria:

Core TWGs The Core TWGs are represented as technical layers in technical architectural models such as the IC's Joint Architecture Reference Model (JARM), DoD's

Defense Information Enterprise Architecture (DIEA), and industry's Open Systems Interconnection (OSI) model.

The Core TWGs are focused on foundational standards that support all of the IT infrastructure that are common among mission areas.

The six Core TWGs are:

1. *Applications*: Manages standards that include programming, coding, web services, and system interoperability. Also covers standards for presentation, user interfaces, databases, and application services. The Applications TWG handles standards in the following OSI model layers: 5 - session layer, 6 - presentation layer and 7 - application layer.
2. *Communications (Comms), Networks, & Physical Connectivity*: Manages standards enabling the support of physical infrastructure and connectivity required to support all network activity. This includes the standards in OSI layers 2 - data link, 3 - network and 4 - transport layers.
3. *Computing, Storage, & Platform*: Manages standards that are responsible for middleware, operating systems, and data storage.
4. *Data*: Manages data standards of common interest across the Enterprise. A data standard is a documented agreement and specification by an authoritative body on a definition, representation, and/or format of data, metadata, and/or exchange protocol that is used to improve data understanding and data interoperability. The Data TWG manages both public data standards such as XML and government data standards developed by IC CIO or other members of the IC. As noted in Section 2.2, under the governance of the JESC, the DSCA acts as the Data TWG. Examples of data standards managed by the Data TWG include various data formats for discovery metadata (e.g., Dublin Core, DDMS, IRM), security marking metadata (e.g., ISM), web and desktop publishing (e.g., Office formats, PDF, PUBS, HTML), semantic data representations (e.g., RDF, OWL, SKOS), and other common content exchanges ubiquitous in the enterprise and not already being addressed by a Domain TWG. This TWG also addresses any data-oriented base technical standards from which a Data Standard was built or profiled (e.g., ASCII, W3C XML family of standards, ASN.1).
5. *Information Assurance/CyberSecurity*: Focuses on Information Assurance standards, policy, and guidance documents that support secure interoperability in net- and data-centric environments, as well as those that support enabling technologies, data architectures, and software tools.
6. *Process Engineering and Management*: Manages standards for configuration management, quality control, mission assurance, quality assurance, planning, integration, and enterprise architecture and engineering.

Domain TWGs The scope of each Domain TWG is delineated by Lines of Business as defined by DoD and/or IC.

The Domain TWGs are focused on IT system standards that are confined to a specific environment that cover all Core areas, e.g., a mission area or line of business.

The Domain TWGs manage standards confined to a specific domain-unique environment that may overlap one or more Core areas, within the scope of the sponsor's authority to ensure consistency in objectives, architecture, and technical approaches for that domain.

The six Domain TWGs are:

1. *Air, Land, Sea, and Space*: Responsible for the interfaces to weapon systems, C4ISR systems, AR systems, aviation systems, radios and radio communications, satellites and tactical messaging and symbology.
2. *Business*: Addresses the exchange of information between organizations and systems, using data standards associated with the business functions of an enterprise, whether it is within the DoD, the Intelligence Community, or any other government or commercial enterprise.
3. *Forensics & Biometrics*: Responsible for the formal adoption and management of IT standards related to the identification of artifacts, evidence, and remains, as well as measurable biological, biographical, contextual, and behavioral characteristics that can be used for the recognition of individuals.
4. *Geospatial Intelligence*: In most instances the TWG's focus is on standards in three Service Areas:
 - GEOINT: Geospatial Standards that include dictionaries, definitions, models, metadata, and formats to facilitate mapping, analysis, exploitation, portrayal, and exchange of geospatial data.
 - GEOINT: Motion Imagery Standards that include the tasking, collection, posting, processing, storage, exploitation, discovery, retrieval, and exchange of motion imagery, associated metadata, audio and other related media types whether generated from electro-optical (EO), infrared (IR), or other motion imagery.
 - GEOINT: Still Imagery Standards that include the tasking, collection, posting, processing, storage, exploitation, discovery, retrieval, and exchange of digital imagery and gridded data associated with geospatial intelligence. The standards in this service area address imagery and gridded data topics such as formatting, compression, support data, metadata, graphical and textual annotations, image quality, and imagery-derived data and products. They are applicable to electrical optical (EO), Infra-red (IR), overhead non-imaging IR (ONIR), synthetic aperture radar (SAR) phase history, data, SAR

(complex and detected) imagery, multispectral imagery (MSI), hyperspectral imagery (HSI), ultraspectral imagery (USI), Polarimetric Imagery (PI), Interferometric Synthetic Aperture Radar (IFSAR), Light Detection and Ranging (LIDAR), raster maps and charts, and Ground Moving Target Indicator (GMTI).

5. *Medical and Health*: A medical (messaging) standard is one that enables clinical applications to exchange radiological, laboratory, hospital, pharmacy, insurance information and electronic health record data, communicating at some or all seven OSI layers. Examples of typical standards are HL7, ASC X12N, and NCPDP. The messaging grew out of the need for EDI communications between organizational systems requiring formalized package content and coding.
6. *Modeling and Simulation*: Manages Information technology standards for models, simulations, and associated data, to standardize the exchange of data between models and simulations, and between models, simulations, and external systems. A model is a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process. A simulation is a model implemented over time. Simulations may be live, virtual, or constructive and may be operated as standalone or in single-architecture and multi-architecture distributed simulation environments.

3.4 - CMSTT and ESTT Specification Management

Changes to existing specifications or the addition of a new specification are submitted via two Change Request (CR) processes. There are two phases in the development of a new or modified specification and entry of the specification into the DISR and the IC ESB through the DISR^[5] tool. The first phase focuses on development of a new specification or changes to an existing specification, along with a technical soundness review of the specification. The second phase results in the addition of the specification into the DISR and IC ESB baselines. Note that a CR in the IC CIO Specification Development Schedule is not the same as a CR submitted in the standards baseline update process. A technical CR submitted through the IC CIO Specification Development Schedule will result in a requirement for a new specification or a technical change to an existing specification. A standards CR submitted through the standards baseline update process will result in a request to update the DISR and IC ESB standards baselines through the DISR^[5] tool.

New technical CRs are discussed at the monthly CMSTT/ESTT meeting. Barring any major push-back from the members, the CR will be worked into the release schedule. If the CR implements requirements from law, Executive Orders, Presidential Policy Directives, other national or community policies, or decisions by chartered IC CIO boards, then the CR will be added to the release schedule regardless of any push-back from CMSTT/ESTT members. As specifications are disseminated for community review, the group of specifications is given a name based on the date that final review is anticipated, for example 2014-DEC.

The review cycle is nominally 4 weeks of initial review by the community, 5 weeks of adjudication, 3 weeks of final review by the community, 3 weeks of adjudication, and 1 week intent to sign for a total of around 16 weeks. Small updates such as a change in values in a vocabulary specification ([Section 5.3.2 - CVE Encoding Specifications \(CES\)](#)) may have a shorter review cycle and may

only have one round of review. Priority updates to specifications may also use a single review process, if the changes are straightforward and can be reviewed successfully within one cycle. Policy-driven vocabulary changes may be published with a zero-review cycle, especially if the vocabulary has a single authoritative source. All comments received during the adjudication process are placed in comment matrices and included in the final review packages. After the technical review cycle has completed, the IC CIO will sign a technical soundness memo which will be included with the final packages.

Once the technical review has completed and the technical soundness memo has been signed, the process enters into the second stage where technical specifications are submitted formally for entry into the DISR and IC ESB baselines. A standards submission CR is submitted into the DISR and IC ESB process through the DISR^[5] tool. In addition to standards developed by the CMSTT/ESTT, CRs may be submitted to add commercial open source, voluntary consensus, or government-developed standards. The standards CRs are reviewed and acted on at least twice a year. Each update is denoted by the last two digits of the year followed by a dash and the number of the update in the year, e.g., 18-3.0 for the third update in 2018. TWG Chairs meet to allocate CRs to appropriate TWGs. The TWG with Primary Responsibility will vote on the CR. In addition, a CR may be assigned to one or more Secondary TWGs, whose members will review the CR and make recommendations to the CR's Primary TWG. Some TWGs hold a formal walk-through session for each to review the CRs assigned to it, followed in about a month with a formal voting session. The results of each TWG's voting session are forwarded to the JESC. The JESC votes on the entire set of CRs across all TWGs. After the JESC votes in Plenary Session, the IC ESB promulgation memo is prepared and signed by the IC CIO and the DISR promulgation memo is prepared and signed by the Department of Defense Chief Information Officer (DOD CIO) and other DOD leadership. The signed promulgation memos are then disseminated to the IC and DOD communities.

Chapter 4 - Specification Conventions

Certain technical and presentation conventions are used in the creation of the IC technical specifications to ensure readability and understanding. These conventions apply to all specifications produced by the IC CIO including this framework.

4.1 - Language

When appearing in all capital letters in a 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 Internet Engineering Task Force (IETF) Request for Comments (RFC) 2119, “Key words for use in RFCs to Indicate Requirement Levels” [19]. When these words appear in regular case, they are meant in their natural-language sense.

4.2 - Terminology

For an implementation to conform to a specific technical specification, it MUST adhere to all normative aspects of the specification. 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.

4.3 - Typography

Certain typography is used throughout the body of a technical specification to convey certain meanings, in particular:

- *Italics* – A title of a referenced work or a specialized or emphasized term
- Underscore – An abstract data element
- **Bo1d Monospace** – An XML element or attribute

4.4 - Dependency Definitions

Specifications often rely on other specifications, components or artifacts, either directly or transitively. 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:

Direct Dependency	Direct influence
	Example: A is influenced by B directly, therefore B is a direct dependency of A.
Transitive Dependency	Transitive influence
	Example: A is influenced by B directly and B is influenced by C directly, therefore C is a transitive dependency of A.

Dependency	Influencer, influences others Example: A influences B, therefore A is a dependency of B.
Inverse Dependency	Influencee, influenced by another Examples: A is influenced by B, therefore A is an inverse dependency of B.

4.4.1 - IC-SF Dependencies

IC-SF.XML is the framework for all other specifications and does not have any dependencies so it does not have a dependency diagram. IC-SF.XML does have inverse dependencies because it is a dependency for all other specifications but it is not a useful diagram to show because it is just a list of all other specifications.

4.4.2 - Dependency Diagrams

The dependency and inverse dependency diagrams will only show IC specification dependencies. It will not show non-IC specification dependencies such as w3 or Taxonomy. Taxonomy is an artifact type for specifications that have need of defining taxonomies and is needed as a package dependency for schema validation.

4.5 - Package Types

In order to better meet the various needs of the community, there are several types of packages included with each release.

4.5.1 - Standalone Packages

The standalone package of a specification contains only specification specific artifacts. It does not include the specifications that it is dependent on (see [Dependency](#)) since there may be more recent versions of those specifications available. In order to obtain all the necessary standalone packages, a specification's dependencies and their dependencies will have to be traversed and obtained. In order for the Standalone package's schema and CVEs to validate and operate as intended, these dependent packages will have to be downloaded and copied into the appropriate directories.

4.5.2 - Convenience Packages

Convenience packages convey all dependencies pre-packaged together and are tested as interoperable. It includes the most recent versions of all dependent (see [Dependency](#)) specifications at the time the package is generated. It is anticipated that a convenience package will be updated when any of its dependent specifications change; however, it will not be signed as a formal package. 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.

Packaging Exceptions:

- **No dependencies**

Specifications that do not have a dependency to another IC CIO specification will not have a Convenience package.

- **Descriptive documents**

Descriptive documents, such as Access Control Encoding Specification (ACES) and Attribute Practice Compliance Statement (APCS), are not XML specifications. While they do have dependencies on other IC CIO specifications, they do not “need” any of them since there are no schema or instance documents of their type. As a result, they will not have a Convenience package.

4.5.3 - Convenience Light Packages

Convenience Light packages contain everything in a single zip that a consumer would require to use the specification, but not everything to make reading it easy. As with the Convenience Packages, the Convenience Light packages convey all dependencies pre-packaged together and are tested as interoperable. It includes the most recent versions of all dependent (see [Dependency](#)) specifications at the time the package is generated. However, in order to make the package smaller in size, the Convenience Light packages do not contain the Schema Guide or any of the dependent PDFs.

Packaging Exceptions:

- **No dependencies**

Specifications that do not have a dependency to another IC CIO specification will not have a Convenience Light package.

- **Descriptive documents**

Descriptive documents, such as ACES and APCS, are not XML specifications. While they do have dependencies on other IC CIO specifications, they do not “need” any of them since there are no schema or instance documents of their type. As a result, they will not have a Convenience Light package.

4.6 - Conformance

For an implementation to conform to a specific technical specification, it MUST adhere to all normative (see [Section 4.2 - Terminology](#)) aspects of the specification.

The XML schemas (unless noted otherwise), CVE values from the XML CVE files, and any Schematron^[29] rules contained in a specification package are normative. The rest of the specification package, including the descriptive content referenced within the XML Schema Guide, the Extensible Stylesheet Language (XSL) transformations, the Schematron Guide, and PDF CVE value files, are informative (see [Section 4.2 - Terminology](#)). Additionally, the use of keywords defined in IETF RFC 2119^[19] is considered normative within the scope of a sentence in the specification document. All other parts 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^[35]. The ability to specify which version of a dependent specification to import enables the configuration change control of parent specifications to be “decoupled” from the configuration change control of dependent specifications. This “decoupling” method has not been in place for all versions of the specifications; therefore, please verify with the dependency table, included in the specification's documentation, 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.

4.7 - Version Policies

4.7.1 - XML Namespace Policy

The XML namespaces defined in a specification do not incorporate a version number and do not change with revisions of the specification. This choice aligns with perspective two from TAG-9-Jan-2006, *The Disposition of Names in an XML Namespace* ^[30]. This decision allows for systems that process information encoded with these specifications to use the same Path Language (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 WEBARCH-15-Dec-2004, *Architecture of the World Wide Web, Volume One* ^[33].

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 a specification are released, the version number captured in the “version” attribute increments. See [Section 4.7.2 - Version Numbering](#) for information on the numbering scheme.

4.7.2 - Version Numbering

The version numbering for a specification is defined by a year-month structure (e.g., YYYY-`MMM`). This provides a temporal representation of when a specification was released. Revisions to a version of a 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 ABNF, *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] RevisionMonth ::= 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. A version is the result of a change, or set of changes, not related to oversights, alignment with law or policy, or break(s) an unacceptable number of existing implementations.
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. A Revision may: modify a version in order to bring the specification into alignment with law or policy; correct failures to fully implement features; correct bugs in the Schema or Schematron that prevent proper usage; correct documentation errors; or update documentation in order to improve understanding.

Examples:

- *202111*: A specification has a Version of 2021-NOV
- *202010.202111*: A specification has a Version of 2020-OCT with a Revision of 2021-NOV
- *202111-CIA_Customization7*: A specification has a Version of 2021-NOV with a customization suffix of CIA_Customization7

- *202010.202111-CIA_Customization7*: A specification has a Version of 2020-OCT with a Revision of 2021-NOV and a customization suffix of CIA_Customization7

4.7.2.1 - Trusted Data Format Versions

The **DESVersion** attributes for TDF specifications are slightly different from all other specifications. In order to account for the foundational BASE-TDF.XML^[4] specification, an additional prefix to the Version attribute was introduced. The Version for a TDF will start with the BASE-TDF.XML^[4] version, followed by the TDF specification version, followed by an optional customization suffix. New versions of BASE-TDF.XML^[4] will always require generating new versions of the derived TDF specifications whereas revisions of BASE-TDF.XML^[4] do not require regenerating the derived TDF specifications.

Examples:

- *202111-CDSM-TDF.202111*: BASE-TDF Version 2021-NOV, CDSM-TDF Version 2021-NOV
- *202111-CDSM-TDF.202111.202205*: BASE-TDF Version 2021-NOV, CDSM-TDF Version 2021-NOV with a Revision of 2022-MAY
- *202111-CDSM-TDF.202111-CIA_Customization7*: BASE-TDF Version 2021-NOV, CDSM-TDF Version 2021-NOV and a customization suffix of CIA_Customization7
- *202111-CDSM-TDF.202111.202205-CIA_Customization7*: BASE-TDF Version 2021-NOV, CDSM-TDF Version of 2021-NOV with a Revision of 2022-MAY and a customization suffix of CIA_Customization7

Chapter 5 - Specification Overview

5.1 - Specification Architecture Approach

ICD 208, *Write for Maximum Utility* ^[14], defines standard methods for producing intelligence that provides the greatest use to customers. The IC CIO data specifications architecture takes a similar approach to defining standards that meet the needs of multiple customers, functions, and operating environments.

Principles found in ICD 208^[14] that also apply to data exchange specifications include:

- Knowing and addressing the needs of different customers and different operating environments
- Developing specifications that support tailored reuse
- Supporting customers' discovery, learning and use of specifications by standardizing specification components, types and families
- Developing different artifacts in a specification and different packages for a specification that support customers at different classification levels and multi-fabric environments
- Applying different specifications development and review cycles that allow rapid delivery of new versions to meet policy changes.

Different IC CIO specifications are targeted toward different functions in the intelligence environment, including authoring and production, discovery, access control, messaging, audit, electronic records, and information protection. Some IC CIO specifications focus on textual products, while others support the exchange of binary data. Some support the exchange of specific types of intelligence data such as data extracted from collected documents and other media. Some specifications support exchange of intelligence data and products, while others enable the exchange of information about persons and non-person entities in support of access control decisions ([Section 5.2 - Components of Access Control Decisions](#)). IC CIO specifications are organized into different categories of specifications to help customers discover and implement the specifications that are applicable to their needs (see [Section 5.3 - Type of Specifications](#) and [Section 5.5 - Families of Specifications](#)).

In addition to functional groupings, there are different types of IC CIO specifications based on their role in information exchange. Some are complex implementations that provide complete support for an information exchange, while other specifications are focused on vocabularies of values used in data elements. Complex implementations are important to software developers building information exchange systems and services. Vocabularies are important to software developers but also to policy makers such as the controllers of the IC Markings, *Intelligence Community Markings System Register and Manual* ^[10], and external standards bodies such as the U.S. Board of Geographic Names (BGN). Still other specifications provide taxonomies that relate one value to another. See [Section 5.3 - Type of Specifications](#).

The different types of specifications also support variable life cycles for development, review and publication of a specification. While full implementation specifications undergo a two-phased review cycle, the simpler vocabulary specifications can be modified, reviewed and baselined in a

rapid single-phase review cycle. This rapid response approach is critical to keeping current with policies that control the values in each vocabulary. Even full, complex implementation specifications can undergo a more rapid, reduced review cycle if the specification version is determined to be a minor revision to an existing specification, under a standardized method of distinguishing revisions from full releases.

Internal to an IC CIO specification are different artifacts that fulfill different purposes in the specification and that meet the needs of different customers. Formal documentation of a specification and controlled vocabularies meet the needs of mission and business customers, while technical artifacts like XML schemas meet the needs of software developers. The different types of artifacts in an IC CIO specification are standardized in a way that facilitates discovery, understanding, use and re-use (see [Section 5.4 - Components of Specifications](#)). Standardization of the types of artifacts in a specification also supports validation that the specification satisfies its requirements.

Some IC CIO specifications are deliberately targeted toward re-use of modular data standards. The Trustable Exchange family of specifications (see [Section 5.5 - Families of Specifications](#)) includes data containers called micro-assertions. These micro-assertions may be community-standard formats supporting general needs such as discovery metadata or electronic records management, or they may be formats developed by communities-of-interest or to support individual elements' or different INTs' needs. The Trustable Exchange wrapper has also been tailored for different purposes, such as exchange of disseminated intelligence products, collected media data and audit data.

A key area of support for different customers and operating environments is seen in the standardized methods for developing, publishing and protecting different packages of a specification at different classification levels. The IC CIO specification architectural framework recognizes the need for specification packages at the lowest possible level of classification for broadest distribution, with more sensitive information published in separate appendices of specification artifacts and combined into alternative packages classified at the appropriate levels. All IC CIO specifications contain classification metadata and portion marks that can be used for automated validation to ensure that a package does not contain files at a higher classification level or with more restrictive control markings than the classification of the overall package.

Where required, some types of specifications may be published in separate specifications, with each specification targeted toward a different security fabric. This approach is used for specifications that support the exchange of identity and access-related data for persons and non-person entities. The generation of different specifications for different security fabrics recognizes both the common and the different requirements for different fabrics, as well as differences in the existing Identity, Credential, and Access Management (ICAM) infrastructure in different fabrics. XML technologies are used to produce one file for common content across fabrics, which is then integrated into the fabric-specific specifications.

5.1.1 - Specification Locations

The IC CIO specifications can be found in several locations:

Top Secret (TS) Intelshare site <https://go.ic.gov/ggDsY2K> (case sensitive – golf golf Delta sierra Yankee 2 Kilo)

Since go.ic.gov Uniform Resource Locator (URL) shortened URLs are not able to be modified after creation we make go.ic's that point to guide numbers. Guide numbers are able to be updated and are the IC standard mechanism for referring to the object.

For example, <https://go.ic.gov/ggDsY2K> resolves to the guide <https://guide.cia.ic.gov/2020/662eddc1-2b01-4134-b2be-1aef4723b3a0> which in turn resolves to <https://intelshare.intelink.ic.gov/sites/odni/cio/csg/Pages/TechSpecs.aspx>

When the Intelshare changes, we can update the guide <https://guide.cia.ic.gov/2020/662eddc1-2b01-4134-b2be-1aef4723b3a0> to point to the new location and the existing go.ic <https://go.ic.gov/ggDsY2K> continues to work.

Unclassified Intelshare Site

Data Specifications: <https://www.w3id.org/ic/standards/Data-Specs>

Service Specifications: <https://www.w3id.org/ic/standards/Service-Specs>

To keep the URL short and to ensure the links remain valid, even if the Intelshare site is reorganized, the redirection service, <https://www.w3id.org>, is used. For example, the above shortened URL will redirect to the actual link, in this case: <https://intelshare.intelink.gov/sites/odni/cio/ea/library/Data%20Specifications/>. If Intelshare reorganized their file structure, <https://www.w3id.org> would update the actual link, leaving the shortened URL unchanged. By using this service, the links referenced in the IC CIO specifications would remain valid.

Unclassified DI2E

<https://subversion.di2e.net/repos/ICSPACKAGES/trunk/>

An account is required and you must be logged in to DI2E before going to this site. To obtain a license, go to <https://www.di2e.net/>.

Public Office of the Director of National Intelligence (ODNI) website

<https://www.w3id.org/ic/standards/public>

To keep the URL short and to ensure the links remain valid, even if the website is reorganized, the redirection service, <https://www.w3id.org>, is used. For example, the above shortened URL will redirect to the actual link, in this case: <https://www.dni.gov/index.php/who-we-are/organizations/ic-cio/ic-cio-related-menus/ic-cio-related-links/ic-technical-specifications>. If Intelshare reorganized their file structure, <https://www.w3id.org> would update the actual link, leaving the

shortened URL unchanged. By using this service, the links referenced in the IC CIO specifications would remain valid.

5.2 - Components of Access Control Decisions

Technical specifications or information guidance documents are used to make access control decisions. Control decisions are based upon three components (data/resource attributes, entity/subject attributes, and access control policies) and are held together by the context in which the access control decision is made. The context itself includes various elements, such as the environment, temporal state, and method of access, that together provide the Where, When, and How details of the access request. The context, together with the user making the request and the data/repository/application being requested (the Who and What respectively), make up the framework that supports an access control decision. Access Policy SHOULD be constrained to use data/resource attributes, entity/subject attributes, and context information. A Policy Decision Point (PDP) uses this framework to make a grant or deny access decision. An entity or subject MUST meet all criteria in the framework to be granted access. The concept of the access control decision framework is depicted in [Figure 2](#).

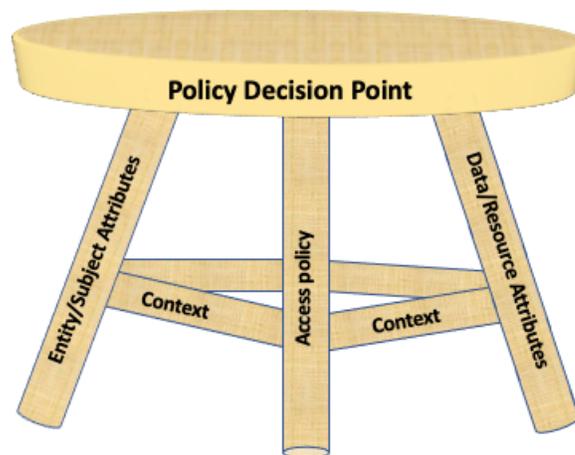


Figure 2 : Three-legged Stool of Access Decisions

All of these parts come together to create a three-legged stool of access control. When a stool is missing one of the components of its frame, it is unable to function properly. The same is true of access control. Without each component of the framework, access control falls apart. Each component is crucial to make accurate, reliable, and automated access control decisions. Each IC CIO document will address a piece of the framework of access control decisions.

Specifications may participate in one or more of the legs of the access control framework either as a primary specification or as a dependency of a primary specification. The primary specifications for the legs include:

- Access policy specifications:
 - *Access Control Encoding Specification for Information Security Markings* (ISM.ACES^[20])
- Data/Resource attribute specifications:
 - *CVE Encoding Specification for Authority Categories* (AUTHCAT.CES^[3])
 - *XML Data Encoding Specification for Information Security Markings* (ISM.XML^[21])
 - *CVE Encoding Specification for ISM Country Codes and Tetragraphs* (ISMCAT.CES^[22])
 - *XML CVE Encoding Specification for License* (LIC.CES^[24])
 - *CVE Encoding Specification for Mission Need* (MN.CES^[25])
 - *CVE Encoding Specification for US Agency Acronyms* (USAgency.CES^[32])
 - *CVE Encoding Specification for Role* (ROLE.CES^[28])
- Entity/Subject attribute specifications:
 - *CVE Encoding Specification for Authority Categories* (AUTHCAT.CES^[3])
 - *CVE Encoding Specification for Fine Access Control* (FAC.CES^[8])
 - *Data Encoding Specification for IC Full Service Directory Schema* (FSD^[9])
 - *Secret Entity Attributes* (IC-SEA.XML^[12])
 - *IC Enterprise Attribute Exchange Between IC Attribute Services Unified Identity Attribute Set* (UIAS.XML^[31])
 - *CVE Encoding Specification for US Agency Acronyms* (USAgency.CES^[32])

Entity/Subject Attribute specifications are primarily intended to be used by an implementer and/or administrator who must configure an Attribute Service to meet the requirements for participation in the Unified Authorization and Attribute Services (UAAS) capabilities. Entity/Subject Attribute specifications define the metadata that is exchanged on person and non-person entities to support ICAM. The audience for Entity/Subject Attribute specifications includes:

- Those responsible for implementing and managing the capabilities that create, provide, modify, store, exchange, search, display, or further process IC enterprise identity attributes.
- Data stewards for protected resources, who will use this information to develop policies for access control.
- Those responsible for provisioning and maintaining UAAS.

5.3 - Type of Specifications

The IC CIO produces various types of specifications that address different and specific enterprise needs and use cases. The specifications cover everything from data and entity attributes to value management and access control. DES, Controlled Vocabulary Enumeration Encoding Specification (CES), Taxonomy Encoding Specification (TES), and Abstract Data Definition (ADD) 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. ACES are primarily intended to be used by those developing tools and services that perform access control decisions. This section describes each type of specification produced and its distinct purpose.

5.3.1 - Data Encoding Specifications (DES)

A DES defines detailed implementation guidance for using XML to encode data. It contains XML Schema and Schematron^[29] descriptions of how to encode a data format. And for its CVEs if any

exists, it contains XSD and RRegular LAnguage for XML Next Generation (RELAX NG) schema fragments for reuse by other specifications, in addition to JavaScript Object Notation (JSON), Comma Separated Value (CSV) and XML formats.

Exceptions:

- ***Data Encoding Specification for IC Full Service Directory Schema (FSD)***

FSD^[9] is a descriptive document and therefore does not contain an XML Schema or Schematron^[29] descriptions of how to encode a data format.

5.3.2 - CVE Encoding Specifications (CES)

The CES specification provides detailed implementation guidance for using XML to encode a CVE. The CES vocabulary defines values used in the transmission of information about a data context. The CES defines the XML elements and attributes, associated structures and relationships, mandatory and cardinality requirements, and permissible values for representing data concepts using XML. And for its CVEs, it contains XSD and RELAX NG schema fragments for reuse by other specifications, in addition to JSON, CSV, XML formats.

5.3.3 - Taxonomy Encoding Specifications (TES)

A TES provides XML taxonomy files for machine processing and human-readable documentation about the mappings between tokens. Taxonomy files are intended to facilitate decomposition and roll-up functions. It provides the relationships between tokens, such as which countries have membership in which Tetragraph.

5.3.4 - Abstract Data Definitions (ADD)

An ADD is an encoding agnostic representation of a concept that is often used to reach business agreement on concepts prior to an actual implementation.

5.3.5 - Access Control Encoding Specifications (ACES)

ACES are rules for mapping entity attributes with data tags to make access control decisions. An ACES codifies the decisional logic for grant/deny access decisions using rules and mappings between data tags and user attributes in a given context.

5.3.6 - Attribute Practice Compliance Statement (APCS)

APCS specifications differ from most other technical specifications in that they are not a specification for producing data formatted in XML. APCS contain English rules that organizations can use to populate Entity Attribute repositories with attributes for Person Entity (PE)s and Non-Person Entity (NPE)s. APCS specifications are document specifications only. There are no schemas or Schematron^[29] rules in APCS specifications.

5.4 - Components of Specifications

5.4.1 - Main Documents and Annexes

The introduction and main prose of a specification (e.g., DES, CES, TES) details the specification's implementation requirements and specific development guidance that goes above and beyond that provided in encoding schemas (e.g., XSD), business constraint rules (i.e. Schematron), and controlled value enumerations. To facilitate the maximum utility of the specifications, one or more annexes may also be created for the specification's main documentation based on sensitivity levels of the information documented.

5.4.2 - Signature Memo

The Signature memo is a scan of the IC CIO's signed, *Technical Soundness Approval of Intelligence Community Technical Specification Updates* memo that authorized the version of the specification. The memo is only included in the Standalone package (see [Section 4.5 - Package Types](#)).

5.4.3 - Controlled Value Enumerations (CVE)

CVEs are a list of values that define the data allowed in an element or attribute. All CVE files use ISM attributes on each value and name to designate the classification of the file and to portion mark each row. In addition to XML, CSV and JSON formats are also provided for each CVE. And for the CVE schema fragments, XSD and RELAX NG formats are provided.

5.4.3.1 - XML Notes

There are XML files provided for all of the CVEs. They are in the CVE folder with the CSV and JSON versions of the information. The CVE folder contains a "CveSchema" folder as a peer to all of the CVE specifications in the package. This folder contains the schema for validating the CVE files themselves. It has no value to an instance document using the CVE specifications.

5.4.3.2 - CSV Notes

There are CSV 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, however, the specifications do not currently use them and 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, may not correctly read UTF-8 special characters. These are found in some country names such as "Republic of Côte d'Ivoire". We added the Byte Order Mark (BOM) as this appears to make newer versions of Excel work properly without the following workaround. If you need to use a CVE that contains such special characters, or you think may contain such characters in Excel, you should:



Note

The following steps tested successfully for macOS Excel version 15.3.9 and Microsoft Windows Excel version 14.0.7; it was unsuccessful for macOS Excel version 14.7.1

1. Open Excel to a blank sheet
2. Under the Data menu choose to get external data from a text file
3. Choose UTF-8 as the file origin
4. Choose delimited as the format
5. Choose next
6. Change from tab to Comma as the delimiter
7. Finish import to get the data in with the UTF-8 Characters properly encoded in Excel

5.4.3.3 - 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, however the specifications do not currently use them and there are no new requirements because of their existence. The JSON files are formatted using JavaScript Object Notation for Linked Data (JSON-LD) based on a proposed method for JSON in National Information Exchange Model (NIEM).

5.4.3.4 - RELAX NG Notes

There are RELAX NG format files provided for all of the CVEs. They are in the Schema folder with the XSD versions of the information. They are provided as a convenience to developers who wish to import IC Specification CVEs into other XML specifications that utilize RELAX NG. They will not affect specifications that do not utilize RELAX NG and there are no new requirements because of their existence. RELAX NG is an alternative schema language for XML and it provides both an XML syntax and a compact non-XML syntax. The XML syntax format fragments are provided with the .rng file name extension and the Compact syntax fragments are provided with the .rnc file name extensions.

5.4.4 - Examples

Example files demonstrate how to implement the elements and attributes of a specification and its relevant dependencies, or how to provide exemplars of useful artifacts using a specification. They are included for convenience and to illustrate valid use of the schema. Most specifications that have a schema or schematron should have examples that are both schema and Schematron^[29] valid to their specification and dependent specifications. CVE specifications, such as ROLE.CES^[28] and ISMCAT.CES^[22], do not have examples. Even though CVE specifications have

Schema files it is not very meaningful to have instances of them. CVEs are components used by other specifications that do have example files. Currently most examples are Publicly Releasable.

5.4.5 - Schema

Schema components define the allowable structure of XML documents to support the specification. Not all specifications include a schema component. Schemas allow for machine validation of the structure of a document. In addition to the schema file, XSD fragments of each of the CVEs used are included. The specifications conform to the NIEM rule that each namespace SHOULD be defined by exactly one reference schema.

There is a special case to mention with respect to TDF specifications in that there are multiple TDF specifications all with the same namespace. This is because TDF XML instance documents can only reference one TDF schema and therefore a single instantiated schema file per namespace in an XML instance document.

Some of these TDF schemas also have "guard" schemas which are created when there is a need for the schema to be passed through a cross domain guard. These "guard" schemas are generated from the TDF schemas and will also have the same namespace however an XML instance document can only use either the regular TDF schema or the "guard" TDF schema.

Overall, the specifications that have schema "guard" files are a special case where they have specific things (e.g. comments, processing instructions, annotations, default `@minOccurs` or `@maxOccurs`, and Information Security Markings (ISM) attributes) stripped down to minimize their memory footprint and make the smallest simple file for a cross domain guard.

All of the schema files use ISM attributes on the schema element to designate the classification of the entire schema file. The Schema attributes are used by the automation to set banner marks in the schema guide. Every element, attribute, type, or group should have annotation/documentation elements giving a definition of the object being created in the schema. These definitions provide the meat of the generated "schemaGuide".

If a specification has a need to define a taxonomy, such as Information Security Marking Country Codes and Tetragraphs (ISMCAT)'s "TetragraphTaxonomy" being a taxonomy that groups countries by tetragraphs, then the Taxonomy schema will be needed for schema validation. Taxonomy itself is not a specification but a type of artifact. It just so happens that the Taxonomy itself has a schema to define the common components of taxonomies and as a result, the Taxonomy folder will appear as a peer to the ISMCAT.CES^[22] schema folder in the package.

5.4.6 - SchemaGuide

The SchemaGuide is the eXtensible Hyper-Text Markup Language (XHTML) Oxygen generated guide to the Schema. The guide is generated from the XSD files and includes element definitions and diagrams of schema components, modified to have classification header and footers. In addition, CSS files are provided and the content modified to use CSS to better display HyperText Markup Language (HTML) elements that were put into annotation/documentation elements. All the "long" file names and image names have been renamed to Universal Unique Identifier (UUID)'s to better accommodate windows Operating System file length issues. Index.html is the starting point.

SchemaGuideSchema is a schema file that is often used to generate the SchemaGuide instead of a specifications default schema whenever there is a need to show additional schema information

beyond a specifications default schema. Some common reasons are for specifications that leverage TDF where it is useful to include the TDF schema as well in the SchemaGuide or for specifications that have multiple schemas that are equally important and does not include a default main schema.

5.4.7 - Constraint Rules (Schematron)

Schematron^[29] is a language for making assertions about the presence or absence of patterns in XML documents. Not all specifications include a Schematron^[29] component. It is a flexible way of expanding or modifying a schema as business rules change and places emphasis on capturing constraints in human language assertions. Schematron^[29] allows author-specified error messages for higher-level error explanations. The Schematron^[29] rules utilize XPath to detect the present or absence of XML patterns, including valid or invalid values from CVEs which are difficult to achieve using grammar-based schema languages. Deleted rules are in the deleted folder and rule numbers are never re-used.

There are two types of assertions in Schematron^[29] called Assert and Report. Asserts enforce that conditions are met and trigger their messages when the test fails. Conversely, the Reports only trigger when the test returns true. Reports are good for informational and debugging messages while asserts are more suited to rule enforcement. The ODNI IC Specifications have chosen to only use Assert since all of our Schematron^[29] is related to rule enforcement. Everything not explicitly disallowed is allowed.

Rules are basically groups of assertions that are evaluated when a specified context is triggered. The context for which rule will trigger is defined in an attribute on the rule element. Within a rule there may be one or more assertions. A rule may also define sch:let elements which are variables that can be used by the assertions. While only the most specific rule for a given context will trigger, in the case of multi-match, all assertions within a rule will be evaluated.

5.4.8 - Taxonomies

Taxonomies are not independent specifications; they are XML files that provide hierarchical relationships between values from CVEs. For example, in ISMCAT.CES^[22] the country-code and tetragraph taxonomy denotes the countries or organizations that make up the membership denoted by a Tetragraph. This may be useful for access control purposes or just simply conveying relationship information between value lists. Taxonomies provide XML files that allow for machine processing and human-readable documentation about the mappings. Taxonomy files are intended to facilitate decomposition and roll-up functions. Taxonomy schemas are provided in order to validate the Taxonomy file.

5.4.9 - XSL

The XSL files are stylesheets that describe how to transform and/or render the XML files that are required by the specification.

5.4.10 - manifest.md5

The manifest.md5 is a generated MD5 checksum of every file in the package excluding the manifest itself. This file is intended to aid users in verifying that the specification package they have is consistent with the officially released package.

5.4.11 - README.xhtml

The “README.xhtml” is a basic HTML file explaining the package contents. Along with a brief description, it contains links to each of the artifacts contained in the package.

5.4.12 - XXXpackage.properties

The “XXXpackage.properties” file contains information about the configuration and SVN status at the time of the build.

5.5 - Families of Specifications

The suite of IC data specifications consists of different categories of specifications. Each category fulfills a specific function that promotes the exchange of information in validated standard formats. Each category consists of a family of multiple specifications with common structures and functions. The different families of IC specifications are described in the following sections.

5.5.1 - Attribute Based Access Control (ABAC) Metadata and Guidance

This family includes the data, entity, and access logic specifications that form the Three-Legged Stool of Access Decisions [Figure 2](#). The data-oriented specifications in this family form the Data Attributes leg; they define information security metadata, to include data access rights and handling requirements, information security markings, and need-to-know requirements and restrictions. The entity specifications in this family form the User/Entity Attributes leg; they define the attributes for representing both person and non-person entities. The access logic specifications in this family form the Access Policy leg; they provide guidance and requirements for evaluating the entity attributes against the information security related metadata in order for PDPs to make positive or negative access control decisions.

5.5.2 - Textual Documents

This family includes specifications containing tagging structures for information resource metadata, mixed textual and media content found in the bodies of publications, source reference citations, classification and control markings, and knowledge assertions. Some of these are used as submission formats for disseminating textual information to the Library of National Intelligence (LNI).

5.5.3 - Trustable Exchange & Micro Assertions

This family includes specifications that either bind arbitrary metadata and resources or assist in the transport of organizational messages and other information assets across enterprise domains. The

specifications in this family include framework formats that contain information payloads as well as extensible micro assertions defined either at the community level or by a community-of-interest. The specifications in this family also include IC standard formats that are micro assertions, such as assertions of source citation metadata, production metadata, revision-recall instructions, and Electronic Records Management (ERM) metadata.

5.5.4 - Descriptive Metadata

This family includes specifications that aid in the definition of data elements, promote information sharing and discovery, or provide electronic library or bibliographic data about information resources.

5.5.5 - Enterprise Audit

This family includes specifications that define audit records and action events.

5.5.6 - Guidance/Infrastructure

This family includes specifications that provide core information about specifications and provide implementation guidance.

5.5.7 - Controlled Vocabulary Enumerations

This family includes specifications that define lists of values paired with their definitions for use in other specifications.

5.5.8 - Abstract Data Definitions (ADDs)

This family includes specifications that define abstract terms for information and metadata. Specifications in other families are generally physical exchange formats, some of which are physical implementations of ADDs.

5.5.9 - Service Security

This family includes specifications that design and implement specific security solutions involving Hypertext Transfer Protocol (HTTP)-based web services that may be implemented using various technologies and approaches (e.g., Simple Object Access Protocol (SOAP) and Representational State Transfer (REST)).

Chapter 6 - Constraints

6.1 - Types of Constraint Rules

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.

6.1.1 - Validation Constraint Rules

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^[29] 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^[29] implementation is informative (see [Section 4.2 - Terminology](#)). Implementers developing alternative validation code should follow the technical rule descriptions and Schematron^[29] logic. Should there be a perception of conflict, implementers should bring it to the attention of the appropriate configuration control body for resolution.

6.1.2 - Rendering Constraint Rules

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.

6.1.3 - "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 Data Element Dictionary concerns. These rules will be expanded and modified as the model matures, and as applicable policies change.

Since constraint rules are only a subset of the entire rule base, an XML document that is compliant with the 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 the rules is not compliant with the authoritative guidance.

6.1.4 - 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.

6.2 - Constraint Terminology

For all specifications that depend on IC-SF.XML, 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.

6.3 - 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.

6.4 - Rule Identifiers

Each constraint rule has an assigned rule identifier, indicated in brackets preceding the constraint rule description. Data validation constraint rule identifiers are prefixed with a specification short name followed by “-ID-” and a 5 digit unique number, assigned from pre-defined ranges to group rules by classification. The numerical ranges are described in [Table 1](#). As the constraint rules are managed over time, IDs from deleted rules will not be reused.

Table 1 - Numerical Rule Identifier Ranges

Rule Identifier Range		Description
Start	End	
00001	09999	Reserved for Unclassified constraint rules

Rule Identifier Range		Description
Start	End	
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

6.5 - Data Validation Constraint Rules

6.5.1 - Purpose

A specification 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.

6.5.2 - Schematron

Schematron^[29] is the formal language used in IC specifications to encode normative data validation constraints. The Schematron^[29] 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^[29] encoding in a specification, and implementers MAY use any encodings, tools, or languages desired to implement validation schemes for conformance to a specification. To conform to a specification, a validator MUST find a document valid *if and only if* the Schematron^[29] implementation by Rick Jelliffe, the editor of the International Organization for Standardization (ISO) Schematron^[29] standard, would find the document valid according to the Schematron^[29] rules in a specification.

For better understanding, the Schematron^[29] rules for a specification may be executed in *Oxygen*[®]^[27] or with an Transformations (XSLT) 2.0-compliant processor using the XSLT 2.0, *XSL Transformations (XSLT) Version 2.0*^[37] transforms in the Schematron^[29] implementation from Mr. Jelliffe.

The constraint rules for a specification are dependent on XPath 2.0^[36] and XSLT 2.0^[37] features. Regarding the use of XPath 2.0 and XSLT 2.0 with Schematron^[29], Mr. Jelliffe stated the following^[23]

By default, Schematron^[29] 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^[29] 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^[37] implementation of Schematron^[29] along with a compiled version of the rules.

6.5.3 - 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. All required elements (and certain conditional elements) **MUST** have content, other than white space¹. Elements, which are allowed to only have text content, **MUST** have text content specified.

6.5.4 - Value Enumeration Constraints

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

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

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

6.5.5 - Additional Constraints

6.5.5.1 - Version Constraints

The specification version is specified through attributes on the root element. The schema constrains the values of these attributes. The version 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. For a DES, the minimum version can be any approved version in the Enterprise Standards Baseline (ESB); however, for a CES, a document **MUST** be validated against the latest known version.

6.5.5.2 - 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“White space” is defined in XML 1.0^[34] as “(white space) consists of one or more space (#x20) characters, carriage returns, line feeds, or tabs.”

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 2](#) 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 2 - Revision Constraints table

Validation Rules Version	11	12	13	13.201701	13.201804	201508	201609
Instance Version							
11	Version Match	Instance Too Old (Tech)					
12	Instance Too New	Version Match	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

6.6 - Data Rendering Constraint Rules

Rendering rules define constraints on the rendering and display of a specification instance documents. The intent is to inform the development of systems capable of rendering or displaying a specification instance data for use by individuals not familiar with the details of the specification 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.

6.7 - Conformance Validation

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

6.7.1 - Schema Validation

For specifications that have no normative (see [Section 4.2 - Terminology](#)) schema, the schema provided with these specifications is an informative aid, and it SHOULD NOT be used for conformance validation.

For specifications that have a normative schema (see [Section 4.2 - Terminology](#)), an instance document of the specifications MUST comply with its specification schema and its dependencies,

and schema validation SHOULD occur prior to other validation steps. If schema validation fails, results from later steps may be indeterminate.



Warning

If *XML Data Encoding Specification for Trusted Data Format* (IC-TDF.XML^[13]) is being used it is critical to follow the validation strategy outlined in IC-TDF.XML^[13] to achieve proper schema validation. Failure to do so will have a high probability of schema invalid data appearing to be valid.

6.7.2 - Business Rule Validation

Validation MUST ensure that instance documents comply with the business rules expressed in a specification.

An instance document MUST comply with the business rules expressed in a specification and those expressed in the specification's dependencies. The business rules in a specification are expressed in Schematron^[29], but it is not necessary for implementers to use the specific Schematron^[29] encoding in a specification. Implementers MAY use any encodings, tools, or languages desired to implement validation schemes for conformance to a specification. To conform to a specification, a validator MUST find a document valid *if and only if* the Schematron^[29] implementation by Mr. Jelliffe would find the document valid according to the Schematron^[29] rules in a specification.

Chapter 7 - Generated Guides

7.1 - Schema Guide

The detailed description and reference documentation for a specification 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 a specification schema as well as an implementation-specific data element dictionary.

The guide was generated with a commercially available product named *oXygen*^[27], 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 “Index.html” with supporting graphics.

7.2 - Schematron Guide

The detailed description and reference documentation for a specification's Schematron^[29] rules can be found in a separate document ending in *_Rules.pdf*, which is located inside the Documentation 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^[29].

Appendix A Change History

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

Table 3 - DES Version Identifier History

Version	Date	Purpose
2019-MAR	March 8, 2019	Initial Release. For details, see Section A.3 - V2019-MAR Initial Release Summary
2021-JAN	January 15, 2021	Routine revision to technical specification. For details of changes, see Section A.2 - V2021-JAN Change Summary
2021-NOV	December 3, 2021	Routine revision to technical specification. For details of changes, see Section A.1 - V2021-NOV Change Summary

A.1 - V2021-NOV Change Summary

Significant drivers for Version 2021-NOV include:

- Community Change Requests

The following table summarizes the changes made to V2021-JAN in developing 2021-NOV.

Table 4 - Data Encoding Specification 2021-NOV Change Summary

#	Change	Artifacts changed	Compatibility Notes
1	Updated the Three-legged Stool of Access Decisions (CR-2019-165). <ul style="list-style-type: none"> • Added "Resource" to the Data Attribute Leg and added Role.CES to the Data/Resource Attribute Leg • Added "Subject" to the Entity Leg 	Documentation	No impact to systems.
2	Updated the DES for better clarity and to remove the CDR sections as they have been retired from the DISR as of the 21-2 baseline. (CR-2021-032).	Documentation	No impact to systems.

A.2 - V2021-JAN Change Summary

Significant drivers for Version V2021-JAN include:

- Community Change Requests

The following table summarizes the changes made to V2019-MAR in developing 2021-JAN.

Table 5 - Data Encoding Specification V2021-JAN Change Summary

#	Change	Artifacts changed	Compatibility Notes
1	Added IC-SF schema and common hash verification schema fragment. (CR-2020-039)	Documentation Schema Schematron IC-SF-ID-00001 added IC-SF-ID-00002 added	Systems using the HashVerification need to be updated to handle the changes.

A.3 - V2019-MAR Initial Release Summary

Significant drivers for Version V2019-MAR include:

- Creation of IC-SF specification.

The following table summarizes the initial release in V2019-MAR.

Table 6 - Data Encoding Specification V2019-MAR Initial Release Summary

#	Change	Artifacts changed	Compatibility Notes
1	Creation of IC-SF specification. (CR-2018-103)	Documentation	Initial Release.

Appendix B Glossary

This appendix lists terms, definitions and sources of the definitions for terms used in this document.

attribute	<p>A distinct characteristic of an object. In the context of ICAM standards for PE and NPE entities, an attribute captures characteristics of PEs and NPEs.</p> <p>Source: ICS 500-30, <i>Enterprise Authorization Attributes: Assignment, Sources, and Use for Attribute-Based Access Control of Resources</i> [18].</p>
Entity	<p>An individual (person), organization, device, or process.</p> <p>Source: NIST 800-56Br1, <i>Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography, Revision 1</i> [26].</p>
Tetragraph	<p>A Tetragraph is a group of countries represented by a four-character string, e.g., FVEY for the Five Eyes countries of USA plus AUS, CAN, GBR, and NZL. Tetragraphs are used in the context of security markings classification, e.g., RELEASABLE TO USA, FVEY.</p>
token	<p>A token datatype is an XML schema language built-in datatype. A token datatype is a string datatype that contains one or more strings separated by a single space, e.g., ism:releasableTo='USA AFG FVEY' is an example of an ISM attribute that has token datatype. A token datatype contains no leading or trailing spaces, no carriage returns, no line feeds and no tab characters. The individual strings in an element or attribute that is a token datatype are referred to as <u>tokens</u>. In the ism:releasableTo='USA AFG FVEY' example, the tokens are 'USA', 'AFG' and 'FVEY'. In contrast, the <u>value</u> of ism:releasableTo is the entire string 'USA AFG FVEY'.</p> <p>Source: https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/#token</p>

Appendix C List of Abbreviations

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

ACES	Access Control Encoding Specification
ADD	Abstract Data Definition
APCS	Attribute Practice Compliance Statement
CES	Controlled Vocabulary Enumeration Encoding Specification
CMSTT	Common Metadata Standards Tiger Team
CSV	Comma Separated Value
CVE	Controlled Vocabulary Enumeration
DES	Data Encoding Specification
DISR	DoD Information Technology Standards Registry
DNI	Director of National Intelligence
DOD	Department of Defense
DOD CIO	Department of Defense Chief Information Officer
DSCA	Data Standards Coordination Activity
ERM	Electronic Records Management
ESB	Enterprise Standards Baseline
ESTT	Entity Standards Tiger Team
FOUO	For Official Use Only
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IC	Intelligence Community
ICAM	Identity, Credential, and Access Management
IC CIO	Intelligence Community Chief Information Officer
ICD	Intelligence Community Directive
IC EA	Intelligence Community Enterprise Architecture
IC ESB	Intelligence Community Enterprise Standards Baseline

ICS	Intelligence Community Standard
IETF	Internet Engineering Task Force
ISM	Information Security Markings
ISMCAT	Information Security Marking Country Codes and Tetragraphs
ISO	International Organization for Standardization
IT	Information Technology
JESC	Joint Enterprise Standards Committee
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
LNI	Library of National Intelligence
NIEM	National Information Exchange Model
NPE	Non-Person Entity
NSS	National Security System
ODNI	Office of the Director of National Intelligence
PDP	Policy Decision Point
PE	Person Entity
RELAX NG	REgular LAnguage for XML Next Generation
REST	Representational State Transfer
RFC	Request for Comments
SOAP	Simple Object Access Protocol
TDF	Trusted Data Format
TDO	Trusted Data Object
TES	Taxonomy Encoding Specification
TS	Top Secret
TWG	Technical Working Group
UAAS	Unified Authorization and Attribute Services
URL	Uniform Resource Locator

UUID	Universal Unique Identifier
XHTML	eXtensible Hyper-Text Markup Language
XML	Extensible Markup Language
XPath	XML Path Language
XSD	XML Schema Definition
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations

Appendix D 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: <https://go.ic.gov/6I5LJNo> (case sensitive – 6 India 5 Lima Juliet November oscar)
Available online Intelink-U at: <https://w3id.org/ic/standards/ADD>
Available online at: <https://w3id.org/ic/standards/public>

[3] AUTHCAT.CES

Office of the Director of National Intelligence. *CVE Encoding Specification for Authority Category (AUTHCAT.CES)*. Available online Intelink-TS at: <https://go.ic.gov/JIMIYN5> (case sensitive – Juliet India Mike lima Yankee November 5)
Available online Intelink-U at: <https://w3id.org/ic/standards/AUTHCAT>
Available online at: <https://w3id.org/ic/standards/public>

[4] BASE-TDF.XML

Office of the Director of National Intelligence. *XML DES Encoding Specification for Trusted Data Format - Base (BASE-TDF.XML)*. Available online Intelink-TS at: <https://go.ic.gov/GC4VEXo> (case sensitive – Golf Charlie 4 Victor Echo Xray oscar)
Available online Intelink-U at: <https://w3id.org/ic/standards/BASE-TDF>
Available online at: <https://w3id.org/ic/standards/public>

[5] DISR

Department of Defense. *DoD Information Technology Standards Registry*. Available online at: <https://gtg.csd.disa.mil/dISR/> continuing on to the actual registry requires a CAC and an account.

[6] DoD Instruction 8310.01

DoD CIO. *Information Technology Standards in the DoD*. 8310.01. 31 July 2017.

31 Jul 2017 edition incorporates Change 1 to the 2 February 2015 edition.

Available online at: <http://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/831001p.pdf>

[7] DoD Instruction 8320.02

Secretary of Defense. *Sharing Data, Information, and Information Technology (IT) Services in the Department of Defense*. 8320.02. 24 June 2020.

24 June 2020 edition incorporates Change 1 to the 5 August 2013 edition.

Available online at: <http://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/832002p.pdf>

[8] FAC.CES

Office of the Director of National Intelligence. *CVE Encoding Specification for Fine Access Control (FAC.CES)*.

Available online Intelink-TS at: <https://go.ic.gov/uZz517T> (case sensitive – uniform Zulu zulu 5 India 7 Tango)

Available online Intelink-U at: <https://w3id.org/ic/standards/FAC>

Available online at: <https://w3id.org/ic/standards/public>

[9] FSD

Office of the Director of National Intelligence. *Data Encoding Specification for IC Full Service Directory Schema (FSD)*.

Available online Intelink-TS at: <https://go.ic.gov/TAHlnW8> (case sensitive – Tango Alpha Hotel lima november Whiskey 8)

Available online Intelink-U at: <https://w3id.org/ic/standards/FSD>

Available online at: <https://w3id.org/ic/standards/public>

[10] IC Markings

Director of National Intelligence (DNI), Special Security Directorate (SSD), Security Markings Program (SMP). *Intelligence Community Markings System Register and Manual*.

Available online Intelink-TS at: <https://go.ic.gov/tGXkwGO> (case sensitive – tango Golf Xray kilo whiskey Golf Oscar)

Available online Intelink-U at: <https://w3id.org/ic/standards/policy/icmarkings>

[11] 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: <https://go.ic.gov/aKlfr9y> (case sensitive – alpha Kilo lima foxtrot romeo 9 yankee)

Available online Intelink-U at: <https://w3id.org/ic/standards/IC-ID>

Available online at: <https://w3id.org/ic/standards/public>

[12] IC-SEA.XML

Office of the Director of National Intelligence. *Secret Entity Attributes (IC-SEA.XML)*.

Available online Intelink-TS at: <https://go.ic.gov/2S2crFZ> (case sensitive – 2 Sierra 2 charlie romeo Foxtrot Zulu)

Available online Intelink-U at: <https://intelshare.intelink.gov/sites/odni/cio/ea/library/Data%20Specifications/ic-sea/default.aspx>

Available online at: <https://w3id.org/ic/standards/public>

[13] IC-TDF.XML

Office of the Director of National Intelligence. *XML Data Encoding Specification for Trusted Data Format (IC-TDF.XML)*.

Available online Intelink-TS at: <https://go.ic.gov/hdwc8fn> (case sensitive – hotel delta whiskey charlie 8 foxtrot november)

Available online Intelink-U at: <https://w3id.org/ic/standards/TDF>

Available online at: <https://w3id.org/ic/standards/public>

[14] 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

[15] 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: <https://go.ic.gov/U7v6ZRL> (case sensitive – Uniform 7 victor 6 Zulu Romeo Lima)

Available online at: http://www.dni.gov/files/documents/ICD/ICD_500.pdf

[16] 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: <https://go.ic.gov/kh8NMVJ> (case sensitive – kilo hotel 8 November Mike Victor Juliet)

Available online Intelink-U at: <https://w3id.org/ic/standards/policy/ICS500-20>

[17] 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: <https://go.ic.gov/OAgmenr> (case sensitive – 0 Alpha golf mike echo november romeo)

Available online Intelink-U at: <https://w3id.org/ic/standards/policy/ICS500-21>

[18] ICS 500-30

Director of National Intelligence Chief Information Officer. *Intelligence Community Enterprise Authorization Attributes: Assignment, Sources, and Use for Attribute-Based Access Control of Resources*. Intelligence Community Standard 500-30. 24 April 2014.

Available online Intelink-TS at: <https://go.ic.gov/lqk775v> (case sensitive – lima quebec kilo 7 7 5 victor)

[19] 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>

[20] ISM.ACES

Office of the Director of National Intelligence. *Access Control Encoding Specification for Information Security Markings (ISM.ACES)*.

Available online Intelink-TS at: <https://go.ic.gov/rOG2Bjt> (case sensitive – romeo Oscar Golf 2 Bravo juliet tango)

Available online Intelink-U at: <https://w3id.org/ic/standards/ISM-ACES>

Available online at: <https://w3id.org/ic/standards/public>

[21] ISM.XML

Office of the Director of National Intelligence. *XML Data Encoding Specification for Information Security Markings (ISM.XML)*.

Available online Intelink-TS at: <https://go.ic.gov/qoNICy7> (case sensitive – quebec oscar November India Charlie yankee 7)

Available online Intelink-U at: <https://w3id.org/ic/standards/ISM>

Available online at: <https://w3id.org/ic/standards/public>

[22] ISMCAT.CES

Office of the Director of National Intelligence. *CVE Encoding Specification for ISM Country Codes and Tetragraphs (ISMCAT.CES)*.

Available online Intelink-TS at: <https://go.ic.gov/mLF5WA9> (case sensitive – mike Lima Foxtrot 5 Whiskey Alpha 9)

Available online Intelink-U at: <https://w3id.org/ic/standards/ISMCAT>

Available online at: <https://w3id.org/ic/standards/public>

[23] Jelliffe

Richard Alan Jelliffe. *FAQ. Frequently Asked Questions: Is Schematron tied to XSLT 1.0?*

Available online at: <http://www.schematron.com>

[24] LIC.CES

Office of the Director of National Intelligence. *XML CVE Encoding Specification for License (LIC.CES)*.

Available online Intelink-TS at: <https://go.ic.gov/lsHgQxJ> (case sensitive – India sierra Hotel golf Quebec xray Juliet)

Available online Intelink-U at: <https://w3id.org/ic/standards/LIC>

Available online at: <https://w3id.org/ic/standards/public>

[25] MN.CES

Office of the Director of National Intelligence. *CVE Encoding Specification for Mission Need (MN.CES)*.

Available online Intelink-TS at: <https://go.ic.gov/ndd7V1R> (case sensitive – november delta delta 7 Victor 1 Romeo)

Available online Intelink-U at: <https://w3id.org/ic/standards/MN>

Available online at: <https://w3id.org/ic/standards/public>

[26] NIST 800-56Br1

National Institute of Standards and Technology. *Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography*. Revision 1. September 2014.

Available online at: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Br1.pdf>

[27] Oxygen

SyncRO Soft. *<oXygen/> XML Editor*.

Available online at: <http://www.oxygenxml.com/>

[28] ROLE.CES

Office of the Director of National Intelligence. *CVE Encoding Specification for Role (ROLE.CES)*.

Available online Intelink-TS at: <https://go.ic.gov/GknYELv> (case sensitive – Golf kilo november Yankee Echo Lima victor)

Available online Intelink-U at: <https://w3id.org/ic/standards/ROLES>

Available online at: <https://w3id.org/ic/standards/public>

[29] 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/>

[30] 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>

[31] UIAS.XML

Office of the Director of National Intelligence. *IC Enterprise Attribute Exchange Between IC Attribute Services Unified Identity Attribute Set (UIAS.XML)*.

Available online Intelink-TS at: <https://go.ic.gov/xQK4AX1> (case sensitive – xray Quebec Kilo 4 Alpha Xray 1)

Available online Intelink-U at: <https://w3id.org/ic/standards/UIAS>

Available online at: <https://w3id.org/ic/standards/public>

[32] USAgency.CES

Office of the Director of National Intelligence. *CVE Encoding Specification for US Agency Acronyms (USAgency.CES)*.

Available online Intelink-TS at: <https://go.ic.gov/wmyIRCV> (case sensitive – whiskey mike yankee India Romeo Charlie Victor)

Available online Intelink-U at: <https://w3id.org/ic/standards/USAgency>

Available online at: <https://w3id.org/ic/standards/public>

[33] 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>

[34] 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>

[35] 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>

[36] 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/>

[37] 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 Director of National Intelligence (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@odni.gov.

Appendix F IC CIO Approval Memo

An IC CIO Approval Memo should accompany this enterprise technical data specification bearing the signature of the IC CIO or an IC CIO-designated official(s). If an IC CIO 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 IC CIO staffing and coordination process leading to signature of the IC CIO Approval Memo. The signature date of the IC CIO Approval Memo will be later than the last modified date shown on the specification artifacts by an indeterminable time period.

Upon signature of the IC CIO 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 ESB as defined in ICS 500-20^[16].