C2 Core Technical FAQ
Version 1.0

Abstract

The purpose of this document is to provide answers to recurring technical questions and concerns about C2 Core version 2.0. The answers in this FAQ represent the consensus of the C2 Core Working Group. These answers are intended for an audience familiar with XML and XML Schema, along with the other standards listed in the glossary.

Status of this Document

This is the final draft of version 1. The findings in this document represent the technical consensus of the C2 Core Working Group. This draft will be delivered to the DSSC co-chairs and posted to the C2 Core Developer Network.

Frequently Asked Questions, with Answers

1. *Does the C2 Core "adapter pattern" cause problems in a C2 Core exchange?*

   There are no known circumstances in which C2 Core's use of adapter elements makes any significant difference in cost, performance, or developer convenience in the implementation of a C2 Core conforming data exchange. The C2 Core adapter for TSPI is not materially different than the way GML elements are wrapped within DDMS. This hasn't caused any problems with the use of DDMS, and it won't cause any problems with C2 Core.

2. *Why doesn't C2 Core use* `xs:import`*? Why doesn't it do "native reuse"?*

   In fact, C2 Core does use the `xs:import` element to bring external schemas into its own schemas through use of the adapter pattern. These external schemas are imported as-is, without any material change. "Native reuse" is not an accepted term in the XML technical community. Different people mean different things by the term, and so the question cannot be easily answered. However, we can observe that C2 Core does import schemas from many other standards (GML, TSPI, ISM, etc.) , and that C2 Core does not require modifying these schemas in any way.

3. *Do the "wildcards" in C2 Core keep C2 Core messages from passing through a cross-domain solution (CDS) guard?*

   No. C2 Core does not require use of the `xs:any` wildcard, which is not allowed in reference schemas. C2 Core does permit `xs:any` in exchange and extension schemas. However, it's always possible to remove those wildcards from the schemas used by the guard, which is the recommended best practice. Nothing required by C2 Core makes it inherently unsafe for CDS guards. It is possible for message designers to create something unsafe, but that is always true whether or not C2 Core is used.

4. *Does C2 Core conform to ISM, NTK, MAT, etc.?*

   C2 Core is not the sort of thing that is able to conform to these standards. Instead, an information exchange designer creates a C2 Core message format using those standards. A software developer then builds an application which produces messages according to that format. Those messages will either conform or not conform, depending on whether the designer and developer have done their work properly. Rather than asking about conformance, perhaps a better question to pose is whether anything in C2 Core makes it difficult or impossible for people to follow the conformance rules for ISM, NTK, MAT, etc.

5. *Does anything in C2 Core make ISM conformance difficult or impossible?*

   No. The schemas for C2 Core import a current version of ISM.XML through a mechanism permitting prompt updates as the ISM specification changes. C2 Core provides flexibility for IES designers by making it *possible* to mark any C2 Core element. It is therefore always possible for C2 Core message designers to create message formats which allow proper markings, and always possible for software developers to create applications which produce properly marked messages.

6. *Does anything in C2 Core make NTK conformance difficult or impossible?*

   No. The schemas for C2 Core do not presently import the NTK schemas. However, it is always possible for C2 Core message designers to import NTK schemas into their IES as needed, in complete conformance to the NTK specification. Future versions of C2 Core may address more direct use of NTK schemas

7. *Does anything in C2 Core make MAT conformance difficult or impossible?*

   Yes. MAT is a specification for taking some information, generating different versions of that information that are releasable to different communities, and wrapping those different versions into a single message. At present MAT is not designed to work with any sort of structured machine-to-machine C2 message – it can't be used with C2 Core messages, or UCore v3.0 messages, for example.

8. *Does C2 Core implement tearlines?*

   C2 Core does not itself implement tearlines. C2 Core allows message designers and software developers to include the markings they need to create tearlines in messages, by permitting ISM attributes wherever they are needed.

9. *Does C2 Core conflict with "safety of navigation"?*

   This term is related to the quality of information products used in navigation: maritime charts, vertical obstruction databases, etc. At present these information products are not distributed as conforming C2 Core messages, so the concern does not apply.

10. *Does C2 Core conflict with GFM DI?*

    The six "Organization Servers" are the authoritative source of force structure data, and will provide that data in GFM-DI format. Some of the facts represented in that data (for example, force management identifiers) may also appear in C2 Core messages. This is not a conflict with the GFM-DI specification or the org server data. C2 Core worked closely with GFM to ensure that it preserves the applicable semantics of GFM information in C2 Core messages.

11. *Does C2 Core need permission to harmonize[1] vocabulary terms from external sources?*

    There is no requirement to obtain such permission in any applicable policy or instruction, nor is this notion widely accepted by the XML technical community. The notion is unworkable. An organization which defines the "eye color" property cannot thereby be given the power to keep all other parts of the enterprise from talking about the color "green". The developers of C2 Core have consulted with subject-matter experts to ensure that applicable semantics are preserved in C2 Core components that are borrowed or harmonized from other vocabularies. The authors of the original vocabularies are not required to use the C2 Core components, but they do not get a veto.

12. *Does C2 Core work with XML compression? Does it require a particular technology?*

    C2 Core exchanges may be employed with or without XML compression. C2 Core does not require use of any particular compression technique or standard.

13. *Does C2 Core support OWL and SKOS?*

    C2 Core does not include any artifact in OWL or SKOS format. However, C2 Core message designers are free to create XML elements and attributes which take values from taxonomies defined in OWL or SKOS documents. This is the support required for the "code/codespace" pattern (see FAQ #14), and is the same support for OWL/SKOS that is found in UCore and GML. Of course, a different meaning for "support" might require a different answer to the question.

14. *Does C2 Core support the "code/codespace" pattern?*

    XML elements following this pattern include a "codespace" attribute to specify the meaning of an identifier or an enumeration value. That value may be supplied in another attribute or in the element content. For example:

```
<gml:identifier codeSpace="http://example.com">ID001</gml:identifier>
<ucore:What ucore:codespace="http://example.com" ucore:code="Thing"/>
```

    C2 Core message designers are free to create XML elements which follow this pattern wherever it is appropriate to their information exchange needs. C2 Core does not require this

[1] Harmonization is a process for modeling, adding, and integrating data components in ways that minimize differences, remove duplication, resolve conflicts, reduce the degree of variation, and achieve consistency across all existing components. [NIEM User Guide, volume 1]

pattern in every element with a value taken from a codelist, because it is not appropriate to every exchange.

15. *Does C2 Core properly define ID attributes for XLink and IDREFs?*

    An optional attribute of type `xsd:ID` is defined for every XML element which conforms to C2 Core. This allows a message producer to insert ID values into an IEP wherever they are needed. (Message designers may make the ID values mandatory wherever they are required.) These ID values are suitable for internal and external references using IDREFs or XLink.

Glossary

DDMS	DoD Discovery Metadata Specification
GFM	Global Force Management
GML	Geography Markup Language
IEP	Information Exchange Package
IES	Information Exchange Specification
ISM	Information Security Marking Metadata
MAT	Multi-Audience Tearline
NTK	Need To Know Metadata
OWL	Web Ontology Language
SKOS	Simple Knowledge Organization System
TSPI	Time-Space-Position Information