



## Implementation Guidelines for Global JXDM

### Introduction

The Global Justice Extensible Markup Language (XML) Data Model (Global JXDM) and Dictionary (Global JXDD) are the result of an effort by the justice and public safety community to produce a set of common, well-defined data elements to be used for data transmissions.

The Global JXDM is a *reference model*. This means it is NOT a rigid standard that must be used exactly as it is in its entirety. The Global JXDM was designed as a core set of building blocks that are used as a consistent baseline for creating exchange documents and transactions within the justice community. While an XML Schema rendering of the entire model exists, it is not a requirement for Global JXDM conformance that this entire schema be used for validation. Nonetheless, there are several informal conformance requirements.

The goal of Global JXDM conformance is for the sender and receiver of information to share a common, unambiguous understanding of the meaning of that information. Conformance to Global JXDM ensures that a basic core set of information (the Global JXDM components) is well understood by the community and carries the same meaning. The result is some level of interoperability that would be unachievable with the proliferation of custom schemas and dictionaries.

### Conformance Rules

This section outlines the informal Global JXDM conformance rules. For more details and references to additional information, refer to subsequent sections.

1. Custom schemas must import and reference the [Global JXDM schema namespace](#) or a correct Global JXDM schema subset (which is the same namespace).
2. If the appropriate component (type, element, or attribute) required for the application exists in Global JXDM, use that component. (i.e., do not create a duplicate of one that already exists).
3. Be semantically consistent. Use Global JXDM components in accordance with their definitions. Do not use a Global JXDM element to represent data other than what its definition describes.
4. Apply XML Schema extension rules correctly and consistently.

### Schema Subsets

The Global JXDD has grown to accommodate the needs of a large and varying user base. Though a large dictionary in itself is not a problem, users can experience difficulties when using the large XML schema generated from the full dictionary. In many practical use cases, only a subset of the full Global JXDD is required. Likewise, it is possible to validate with a reduced set (a subset) of the Global JXDM components. For a detailed explanation of the concept of schema subsets, refer to [Customized Schemas](#). To learn how to construct subsets manually, refer to [Rules for Schema Subsets](#).

Realizing that the rules for manually creating a Global JXDM schema subset can be daunting and potentially error-prone, an online tool that can automatically generate a correct schema subset has been developed. This tool, the Schema Subset Generation Tool (SSGT), is currently undergoing beta testing. In the interim, the [GJXDM Viewer](#) is available for searching and navigating the model and the content of the dictionary. In the near future, the SSGT will subsume the Global JXDM Viewer capabilities.

## Reference Architecture

In order to understand how to use a Global JXDM schema subset for validation, it is essential to understand the Global JXDM *reference architecture*. A very brief introduction to this concept follows.

In general practice, an XML instance references an XML schema, which in turn, references the World Wide Web Consortium (W3C) XML Schema specification (by namespace). The instance is valid if it conforms to its respective schema definition, which then, conforms to the W3C XML Schema specification. The Global JXDM schema rendered from the Global JXDM introduces a set of types, elements, and attributes as predefined building blocks for use in justice schemas. Within the Global JXDM schema, these components are optional, over-inclusive, and unconstrained. However, for practical use in validating real instances, the correct components must be identified and constrained as necessary. This process requires validation against two slightly different schemas: (1) the full Global JXDM schema or a correct subset, and (2) a constraint schema that reflects the subset but with applied constraints. The two schemas represent two distinctly different validation paths. The first validates for conformance to the Global JXDM, and the second validates for conformance against the user's required constraints.

In review, instead of:

- an instance,
- the user schema, and
- the W3C Schema specification,

the result is:

- an instance,
- the user schema,
- an optional user extension schema,
- the full Global JXDM schema or a Global JXDM schema subset,
- a corresponding constraint schema, and
- the W3C XML Schema specification.

For a brief illustration of these concepts refer to the presentation on the [Reference Architecture](#) page. For more detailed examples and discussion, refer to the presentations and online training materials from the [Global JXDM Developer's Workshop](#) held in Atlanta, Georgia.

## Component Extension

There are several ways that a local schema might extend the Global JXDM. A simple set of examples has been prepared that illustrates various [extension methods](#). One of these methods is based on the World Wide Web Consortium's (W3C) rules for [extension of XML Schema types](#). W3C Schema rules for type extension allow many possibilities. However, type extension within the Global JXDM is intended to maintain a class hierarchy of objects by adhering to a more restrictive set of subclassing rules.

To ensure the integrity, consistency, and meaning of the Global JXDM class (inheritance) hierarchy, the following rules for type extension must be followed:

1. A derived type may add (extension) additional fields elements/attributes) to its base type.
2. A derived type may restrict one or more fields of its base type, but only so that a derived field is a subset of the field of the base type.  
For example, a derived type may:
  - Restrict an enumeration from a large set of options to a smaller set of options, as long as every option in the derived set appears in the base set.
  - Remove a field of the base type only if the field is optional in the base type.
  - Require a field to appear only if the field is optional or required to appear in the base type.
3. A derived type may not modify a field of its base type such that it violates the constraints of its base type.  
For example, a derived type may NOT:
  - Add additional enumerations to a field.
  - Remove a field that is required by its base type.
  - Modify the type of a field of its base type.

## **Additional Remarks About Conformance**

Exchanges conform to the Global JXDM, systems do not. The way data is labeled or used in one system does not impact Global JXDM conformance. Conformance relies upon how data is packaged as XML for an information exchange.

There is no concept of partial conformance. Just because you use parts of the Global JXDM to exchange with other justice agencies does NOT mean you are conformant. (You may be violating Rule 2.)

These conformance rules are, by design, non-rigid. More formally specified rules would be counter-productive to the development of the evolving Global JXDM.