

August 18, 2014

Draft HMIS XML Data Exchange Specifications

2014 HMIS Data Standards



OFFICE OF COMMUNITY PLANNING AND DEVELOPMENT

HUD HMIS XML 4.0 Documentation

08/18/2014, version: 4.0.0-alpha.6

*This document was developed for the
US Department of Housing and Urban Development
under a contract with Cloudburst Consulting Group.*

Primary authors are:

*Eric Jahn, Alexandria Consulting LLC and
Molly McEvelley, Abt Associates*

[Introduction](#)

[Audience for This Document](#)

[What Is This Document About?](#)

[Purpose of the XML](#)

[Alternatives](#)

[Where to Find the HMIS XML Schema](#)

[Components](#)

[Overall Structure](#)

[Figure 1: HMIS XML 4.0 Basic Structure](#)

[Inline Documentation](#)

[Browseable Graphical Documentation](#)

[Relationship to Logical Model](#)

[Cardinality Enforcement](#)

[Changes from Previous Version](#)

[Technology](#)

[XML Schema 1.1](#)

[Extensibility](#)

[Keys](#)

[Examples](#)

[Example Instance](#)

[Example Custom Extension](#)

[Change Process](#)

[HMIS XML Schema Hosting](#)

[Further Information](#)

Changes with this version:

- added clarifications under Overall Structure
- added discussion of cardinality enforcement now with assertions
- added deletion handling discussion
- added hashing discussion

Introduction

Audience for This Document

This document is intended for people implementing software systems which need to transmit HUD HMIS project and client information over a network. Because of this, it is inherently a technical document. Those new to eXtensible Markup Language (XML) should first familiarize themselves somewhat with that technology in general, before examining the [actual schema product](#) and the [Technology](#) sections.

What Is This Document About?

The HUD HMIS XML Schema, version 4.0 is a format for transferring the HMIS data. The data elements are defined in the [May 2014 HUD HMIS Data Standards Manual](#), and further specified by the [May 2014 HUD HMIS Data Dictionary](#) and [2014 HMIS Logical Model](#). XML is validatable and extensible. This means that you can independently and automatically check if this data is being sent to specifications, which is XML's main benefit. You can customize it also, but still keep the ability to check it for accuracy. This format can be used for data migrations between systems, or the data types defined within it could be individually referenced in custom (not officially HUD specified) API methods.

Purpose of the XML

This schema requires complete data sets for each group of data (groups like Inventory, Project, Path Status, etc.), for primarily reporting purposes. Because of this, most of the data elements are not optional. For example, if you wish to include any of the Inventory data element within the Logical Model, you must include all of the data elements. In past versions of the HMIS XML Schema (v3.1 and prior), most data elements were listed as optional, to accommodate use cases other than reporting complete data sets.

Alternatives

For HMIS implementers seeking an alternative format for HMIS client data transmission, see the HUD Comma-Separated Values (CSV) for HMIS data, version 4.0.

Where to Find the HMIS XML Schema

The HMIS XML 4.0 and previous major releases are hosted at <http://www.hudhdx.info/VendorResources.aspx>

Components

This publication has multiple parts, each is available in two places: the [HUD HDX link](#) above, and a [development website](#), where new versions are discussed, and all changes are stored.

- This document, which includes:
 - A rationale for the schema, including an overview of the process, and an explanation of the model.
 - A description of the steps involved beyond creation of a data standard, including development of communication protocols and documentation of responsibilities.
 - A brief discussion of the future path of HMIS XSD development.
- An XML Schema Definition (XSD) document: [HMIS XSD v4.0](#) (temp location). It contains inline documentation which correlates each schema element to an item in the HMIS Data Dictionary and Data Manual. Searching the HMIS XML Schema for the corresponding HMIS Data Dictionary data element numeral provides a cross-walk between technical schema and the HMIS Data Standards.
- A sample, valid XML document with fictitious data.
- An example extension schema of the HMIS XSD 4.0, illustrating how to add an additional data element.

- A sample, valid XML instance document for the extended schema.
- Online, browseable, [graphical documentation](#) (temp location) for version 4.0

Overall Structure

The HMIS XML 4.0 structure is intended to be completely compliant with its parent specification, the HUD HMIS Logical Model, and with its other similarly purposed product, the HUD HMIS CSV 4.0. To keep it simple, the XML has a flat, single file structure. Almost all the data types are one level below the Export data element and key references enforce relationships between types. This allows the system serializing the XML to not have to worry about being in the correct context to add elements. They can be added in any order within the “Export” element.

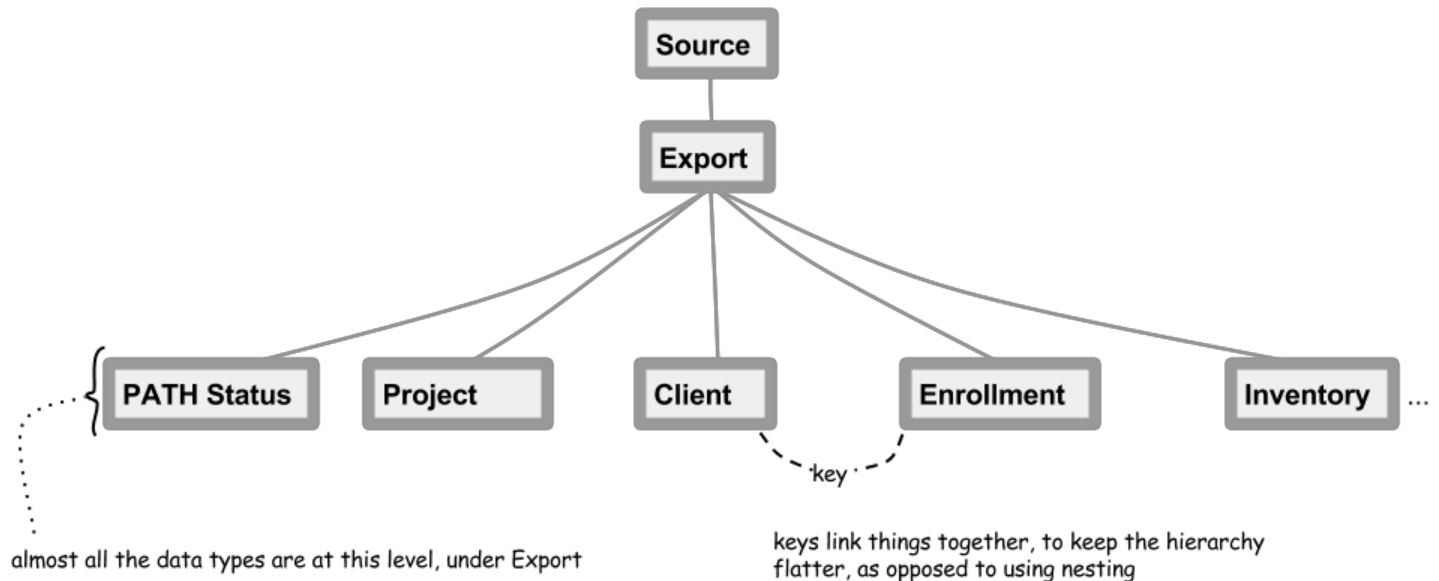


Figure 1: HMIS XML 4.0 Basic Structure

Inline Documentation

The HMIS Data Dictionary and Data Manual data element reference is listed within each HMIS XML data element, whenever available. Some data elements, such as Export, do not reference a data element in the Dictionary/Manual, since they only serve a functional purpose for data transfer.

Browseable Graphical Documentation

0..∞	Client	Type	hmis:client	⊖
0..∞	ClientVeteranInfo	Type	hmis:clientVeteranInfo	⊖
	See May 2014 HMIS Data Dictionary, Section 4.41 See May 2014 HMIS Data Manual, Section 4.41.			
0..∞	CoC	Type	hmis:coc	⊖
0..∞	Contact	Type	hmis:contact	⊖
0..∞	DateOfEngagement	Type	hmis:dateOfEngagement	⊖

A browsable list of all the data element in the HMIS XML Schema is available at [here](#) (temporary location, to be moved to hudhdx site). The definitions for each data element are available in the browseable documentation as well.

Relationship to Logical Model

The HMIS XML Schema version 4.0 complies with [Logical Model](#) (temp location), The various relationship lines within the Logical Model are enforced by requiring matching IDs/Keys (see the section on [Keys](#)) between the two data types (like ProjectCoC to Inventory, or Exit to Enrollment) joined by a line in the Logical Model.

Cardinality Enforcement

The Logical Model is also very specific about cardinalities (like “many-to-many” and “zero to one”) between the related data types. HMIS XML Schema strictly enforces cardinality by means of XPath 2.0 assertions¹; a new feature of XML Schema 1.1. If the Logical Model requires “zero to one” of something (like Date of Engagements per Enrollment), the HMIS XML 4.0 will likewise only let you declare the required zero or one of them.

Changes from Previous Version

- All data elements updated from the 2010 to the 2014 HUD HMIS Data Standard elements
- AIRS namespace elements are dropped, so that the schema is purely defined by the scope of the HUD HMIS May 2014 Data Manual and Data Dictionary
- Line-by-line comparison of version 3.1 to version 4.0.0-alpha.6:
<https://github.com/hmis-interop/xml/compare/v3.1...v4.0.0-alpha.6>

Technology

XML Schema 1.1

[XML Schema version 1.1](#) is used by the HMIS Schema 4.0. XML Schema 1.1 is a superset of XML Schema 1.0 used in previous versions of the HMIS XML format, and is forward compatible from XML Schema 1.0. XML Schema 1.1 simply adds additional features to those of 1.0, two of which are used by HMIS XML 4.0. These are extensibility features, and better key reference enforcement. All major XML parsers have been updated to support version Schema 1.1, which was released in 2007, so XML Schema 1.1 is a mature technology.

Extensibility

Flexible vendor extensions are much more easily accomplished with XML Schema 1.1, as opposed to the previous 1.0. To extend the HMIS XML Schema 4.0 using this new feature, please read the [documentation](#) on *defaultOpenContent mode="interleave"* at the The World Wide Web Consortium (W3C) website. In a nutshell, it permits you to add new data elements anywhere in the HMIS XML Schema, and the original parts still can be found and validated. The newly added extensions which a software provider might add should be validated by a second XML Schema which they can define elsewhere, and both the HMIS Schema and the new extension schema can work together to validate the extended XML documents. The extended schema will still validate against the official HUD HMIS XML 4.0, but it will not complain about the custom elements added.

Currently, HUD HMIS is not publishing any extensions, but the parties engaged in data integration may negotiate the use of an extension schema to be overlayed onto the HUD HMIS XML Schema for the addition of new data types, attributes, and elements.

¹ <http://www.w3.org/TR/xmlschema11-1/#cAssertions>

One potential extension that has been identified is the ability to indicate whether a field value is hashed or not, for deidentification/warehousing purposes. Indicating and transmitting hashing was a capability in prior versions of the HUD HMIS XML Schema. But hashing is seen as a more implementation dependent feature, so hashing has been removed from the current HMIS XML Schema. However, a suggested hashed value model could include a "hashValue" attribute added for certain schema elements, such as personal identifiers. hashValue would contain the hash value, and the element it applies to could have an overridden, dummy value that still validates. This model would make it so you could, say, enforce that the value of DateOfBirth is still a date (albeit a dummy one), yet also transmit the hashing in the "hashValue" attribute. It also makes it easy extend the hashValue attribute to other elements that haven't traditionally included it (like Address).²

Keys

The HMIS XML schema uses keys and references to those keys (aka "keyrefs") to create the relationships defined in the Logical Model. The keys are enforced, so that if an enrollment in the XML references a project ID which isn't defined somewhere else in the XML, it will raise a validation error, which enforces complete, self-referential data sets. Keys also let us flatten the schema. The advantage of a flat schema is data elements don't need to be repeated within deeply nested structures. Keys also makes programming easier, since there are not so many nested logic structures to handle. It is also more flexible, since not every related data element has to be mentioned within the same tree branch of the XML.

Examples

Example Instance

A fairly exhaustive example HMIS XML instance is available [in the repository for download](#) (temp location).

Example Custom Extension

(may be available in a future version of this document)

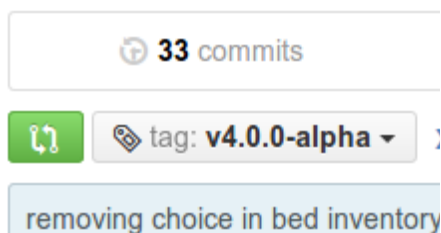
Change Process

To request changes to the HUD HMIS XML Schema, there are multiple ways to register your requests.

- A software issue tracker is available at: <https://github.com/hmis-interop/xml/issues>
- A HUD sponsored [HMIS Software Provider forum](#) regularly hosts a revision process to maintain the HMIS XML Schema
- Email hmisdxc@gmail.com (temporary address) to discuss your request with a HUD Technical Assistance representative

HMIS XML Schema Hosting

HMIS XML schema development archives and issues list is unofficially (not an official HUD Website) hosted at: <https://github.com/hmis-interop/xml>. The tags on that site contain the various versions available, from which the differences between versions can be listed in detail.



² <https://github.com/hmis-interop/xml/issues/27>

Further Information

Contact the [OneCPD Ask A Question](#) to request general assistance with the HMIS XML Schema. For direct technical recommendations or questions send an email to a temporary email set up at hmisdx@gmail.com (temp address), or if you wish to log a detailed change directly, an [issue tracker](#) is available.