

HMIS XML v5.1 Data Exchange Format Specifications

for HMIS Data Standards Version 5.1, effective October 1, 2016



OFFICE OF COMMUNITY PLANNING AND DEVELOPMENT

HUD HMIS XML Schema Documentation

1/20/2016, version: 5.1.1

This document was developed for the US Department of Housing and Urban Development (HUD), Office of Special Needs Assistance Programs (SNAPS) in the Office of Community Planning and Development, under a contract with Abt Associates, Inc..

Primary author:

Eric Jahn, Alexandria Consulting

Contributing authors:

Molly McEvelley, Abt Associates

Table of Contents

Overview.....	1
Audience for this Document.....	1
Purpose of this Document.....	1
Alternative.....	1
Use Cases and Extension.....	1
Scope of the HMIS XML Schema.....	1
Location of the HMIS XML Schema.....	1
Components.....	1
Overall Structure.....	1
Figure 1: HMIS XML 5.1 Basic Structure.....	1
Inline Documentation.....	1
Browsable Graphical Documentation.....	1
Relationship to Logical Model.....	1
Cardinality Enforcement.....	1
Changes from Previous Version.....	1
Technology Overview.....	1
XML Schema 1.1.....	1
Extensibility.....	1
Hashing.....	1
Keys.....	1
Conditional Validation.....	1
Examples.....	1
Example Instances.....	1
Example Custom Extension and Instance.....	1
Change Process.....	1
HMIS XML Schema Hosting.....	1
Further Information.....	1
Overview of Changes.....	1
Change List.....	1
Bugs Fixes.....	1

Document changes with this version are available in the [online versioning system for version 5.1](#).

Overview

Audience for this Document

This document is intended for developers and project managers 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 with XML technology in general, before examining the [actual schema product](#) and the “[Technology Used](#)” section of this document.

Purpose of this Document

The US Department of Housing and Urban Development (HUD), in cooperation with the Department of Health and Human Services (HHS) and the Department of Veterans Affairs (VA), is responsible for the Homeless Management Information System (HMIS) Data Standards, which define data collection requirements for any software used as an HMIS. The first version of the data standards was published in 2004; the current version (v5.1) has an effective date of October 1, 2016.

The HUD HMIS XML Schema, version 5.1 is a format for transferring this HMIS data. Please note that the XML Schema version numbering is now the same as the HMIS Data Standard version numbering. The data elements are defined in the [August 2016 HUD HMIS Data Manual](#), and further specified by the [August 2016 HUD HMIS Data Dictionary](#) and [v5.1 HMIS Logical Model](#). The Logical Model, Manual, and Dictionary all changed between 2014 v3.0 and 2014 v5.1, and so the XML Schema has been changed accordingly. The key benefit of XML is its ability to be validated and extended. This means that systems both sending and receiving data can independently and automatically check if this data is being sent to specifications. This helps to evaluate data quality and correctness independently. The HMIS XML Schema can be customized, as well, but still keep the ability to check the generated data for accuracy against this original HMIS XML Schema. This XML format can be used for data migrations between systems, or the data types defined within it could be individually referenced in custom web methods.

Alternative

For HMIS implementers seeking an alternative format for HMIS client data transmission, see the HUD Comma-Separated Values (CSV) for HMIS data, version 5.1¹. Both the XML and CSV sister formats are based off the same [2014 HMIS Logical Model v5.1](#) (released September 2016, updated January 2017), and are completely semantically and logically compatible because of this fact.

Use Cases and Extension

There are a number of purposes for which HMIS data might be exported from one system and imported to another. The use cases that the HMIS CSV and HMIS XML formats are primarily intended to support include migration from one HMIS application to another; warehousing of data from multiple HMIS implementations for analysis and reporting; and participation in a local HMIS implementation by regularly providing data entered into and exported from an alternate database. In general, HUD expects that it should be possible to export, in a standard format, all data entered into an HMIS for any data element defined by the HMIS Data Dictionary, regardless of whether or not a given data element is required based on project type or funder. Specific requirements related to the overall functionality of HMIS applications and export and import processes in particular will be established by the forthcoming HMIS Software and Data Quality Standards and may result in revisions to this document. The HMIS Data Dictionary defines several hundred fields. HUD is aware that there is, in addition, a wide variety of expanded and customized data collection in HMIS implementations across the country. While HUD is cognizant that the exchange of additional data that may be included in an HMIS will often be useful and necessary, it is not practical to include accommodations for every potential need, even if it

¹ http://www.hudhdx.info/Resources/Vendors/5_1/HMIS_CSV_Specifications_5_1.pdf

were possible to anticipate them. As a result, the scope of this document is generally limited to data collected in a manner consistent with the HMIS Data Dictionary. The HMIS XML format may be extended to include additional fields by parties engaged in HMIS data exchange; technical assistance may be available. To request technical assistance please visit www.hudexchange.info.

Scope of the HMIS XML Schema

This schema requires complete data sets for each grouping of data (groups like Inventory, Project, Path Status, etc.), to accommodate the use case of transmitting complete data sets for reporting, periodic bulk data uploads from a partner agency to an HMIS or data warehouse, or for HMIS system migration purposes. Because of this intended use case, many of the data elements are not optional. For example, if an XML document following the Schema includes any of the data elements within the “Funder” class of the Logical Model, it must include almost all of the required data elements within Funder (since for Funder, all its elements are required, except one). In past versions of the HMIS XML Schema (prior to v4.0), most data elements were listed as optional, to accommodate use cases other than reporting complete data sets. For other use cases, the individual data groups could be individually transmitted in bulk (by removing the key and cardinality restrictions/assertions within the Export element), but that is beyond the scope of this current specification. It should be possible to export, in a standard format, all data entered into an HMIS for any data element defined by the HMIS Data Dictionary, regardless of whether or not a given data element is required based on project type or funder.

Location of the HMIS XML Schema

The HMIS XML Schema version 5.1 and previous major releases are hosted at <http://www.hudhdx.info/VendorResources.aspx> . They are also kept [in a versioning system](#).

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, issues logged, and all incremental 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 version 5.1](#). 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. A [minimal instance](#) is also available there for testing.
- An [example extension schema](#) of the HMIS XSD version 5.1, illustrating how to add an additional data element.
- A [sample, valid XML instance document](#) for the extended HMIS XSD version 5.1 schema.
- Online, browsable, [graphical documentation](#) for HMIS XSD version 5.1.

Overall Structure

The HMIS XML 5.1 structure is intended to be completely compliant with its parent specification, the 2014 v5.1 HUD HMIS Logical Model (released September 2016), and with its other similarly purposed product, the HUD HMIS CSV 5.1. To keep its structure simple, the XML continues to be comprised of a relatively flat, single file. 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 be constrained by having to be in the correct context to add elements. Elements can be added in any order within the “Export” element.

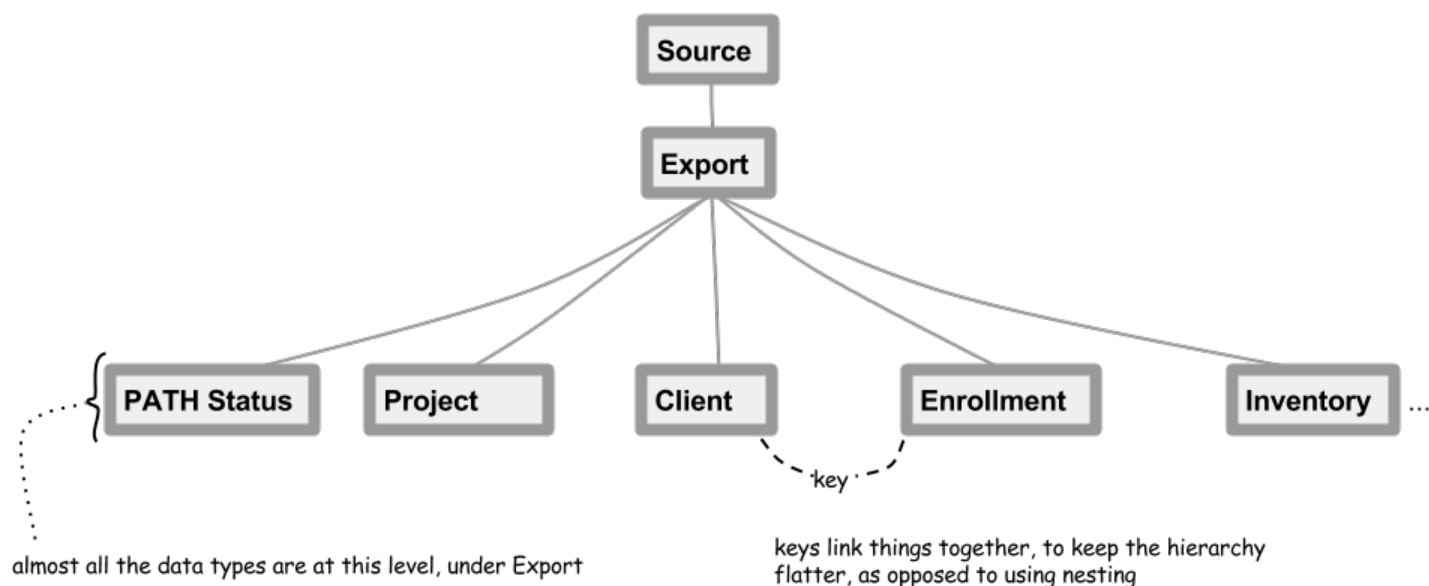
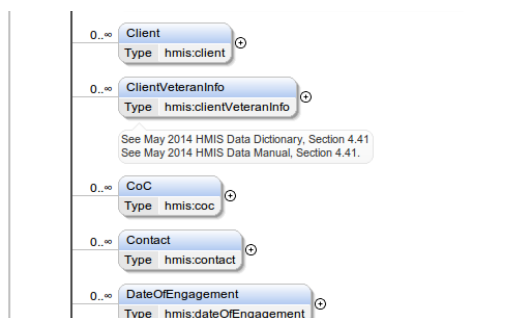


Figure 1: HMIS XML 5.1 Basic Structure

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.

Browsable Graphical Documentation



A browsable list of all the data element in the HMIS XML Schema is [available](#). The definitions for each data element are available in the browsable documentation as well.

Relationship to Logical Model

The HMIS XML Schema version 5.1 complies with [Logical Model](#) (temporary location until posted on HUD HDX), 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 CoC 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 assertions²; 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 5.1 will likewise require valid XML documents to declare the required zero or one occurrence.

Changes from Previous Version

- All data elements are updated from the HUD HMIS Data Standard 2014 version 3.1 (released in September 2015) to the HUD HMIS Data Standard 2014 version 5.1 (released August 2016) elements.
- Line-by-line comparison of version 4.1-latest to version 5.1-latest: <https://github.com/hmis-interop/xml/compare/v4.1-latest...v5.1-latest> (this will stay updated with later big-fixes). For a less concise list of changes, see the [Overview of Changes](#) below.
- Conditional validation of newly added dependent fields. See the section below on [Conditional Validation](#).

Technology Overview

XML Schema 1.1

[XML Schema version 1.1](#) is the format and standard used by the HMIS XML Schema 5.1. It was first introduced with HMIS XML Schema 4.0. XML Schema 1.1 is a superset of the XML Schema 1.0, used in version 3.1 and below of the HMIS XML format, and is forward compatible from XML Schema 1.0. XML Schema 1.1 only adds additional features to those of 1.0, three of which are used by HMIS XML 5.1. These are extensibility features, conditional validation, and better key reference enforcement features. All major XML parsers have been updated to support version Schema 1.1, which was released in 2007, resulting in XML Schema 1.1 now being a mature technology. A practical example of command line validation using the Xerces XML Parser is in the [wiki](#).

Extensibility

Flexible vendor extensions are much more easily accomplished with XML Schema 1.1, as opposed to the previous XML Schema 1.0. To extend the HMIS XML Schema 5.1 using this new feature, please read the [documentation](#) on `defaultOpenContent mode="interleave"`, `xs:redefine`, and `xs:override` and at the The World Wide Web Consortium (W3C) website. This new feature permits is implemented in HMIS XML, allowing the addition of 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. The second XML Schema can be defined elsewhere, and both the HMIS Schema and the new extension schema can function in concert to validate the extended XML documents. The extended schema will still validate against the official HUD HMIS XML 5.1, but it will not invalidate the custom elements added.

Currently, HUD HMIS is not publishing any official 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.

Hashing

Hashing is possible for personal identifiers. There are three possible values for the “hashStatus” attribute:

² <http://www.w3.org/TR/xmlschema11-1/#cAssertions>

- “Unhashed” - Data are expected conform to data type and field length parameters defined in the schema.
- “SHA-1 RHY” - The SHA-1 algorithm produces a 40-character string of letters and numbers. More specifically, for the following identifiers, a specific hashing method is required:
 - FirstName – SHA-1 hash of the SOUNDEX of the value for first name;
 - MiddleName (S40) – SHA-1 hash of the SOUNDEX of the value for middle name;
 - LastName (S40) – SHA-1 hash of the SOUNDEX of the value for last name; and
 - SSN (S44) – concatenation of the unhashed last 4 digits of the SSN followed by SHA-1 hash of the full SSN.
- “Other” - Identifies exports in which data are hashed based on parameters mutually agreed upon by the sender and recipient of the XML data.

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 enable the flattened structure of the schema. An advantage of a flattened schema is that data elements are not repeated within deeply nested structures. Keys also makes programming simpler, since there are not so many nested logic structures to handle. The flattened structure is also more flexible, since not every related data element has to be mentioned within the same tree branch of the XML.

Conditional Validation

The ability to check for a specific precondition before validating is employed in HMIS XML Schema 5.1. The main use is checking first to see if a previous question has been answered in a specific way before allowing a follow-up question to be asked. As with the CSV specification, dependent fields will be null unless the parent field contains a value defined by the HMIS Data Dictionary as triggering the dependent field. Conditional validation is only applied for most newly added questions introduced in version 5.1 (2016) of the HUD HMIS Data Standard, so their coverage of all possible conditional logic is not comprehensive. The main purpose of this conditional validation is to catch typical problems, and to enforce some of the most clearly described question workflows in the Data Standard. Conditional logic is implemented using XPath 2.0 assertions, similar to the existing key reference and cardinality enforcement assertions. This conditional validation can slow the validation time for large XML files, so it can be turned off by simply commenting out the various lines in the 5.1 Schema beginning with “assert”. A regular expression run to remove or comment out all lines in a schema that begin with `<xs:assert>` would accomplish this task, and allow for “primary validation” which excludes the more time consuming “secondary validation” with conditional assertions. A good testing approach is to test small development XML instance files first with secondary conditional assertion validation, and then use the primary validation for fast production validation of large files.

Here is an example of conditional validation in the schema (at [line 278](#)), using an assertion:

```
<!--Testing if DisabilityType is 8, before allowing use of TcellCountAvailable-->
<xs:assert test="if (hmis:TCellCountAvailable) then boolean(hmis:DisabilityType/text()='8') else true()"/>
```

Examples

Example Instances

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

Example Custom Extension and Instance

An example HMIS XML extension is available [in the repository for download](#). An instance applying this extension is also [in the repository for download](#).

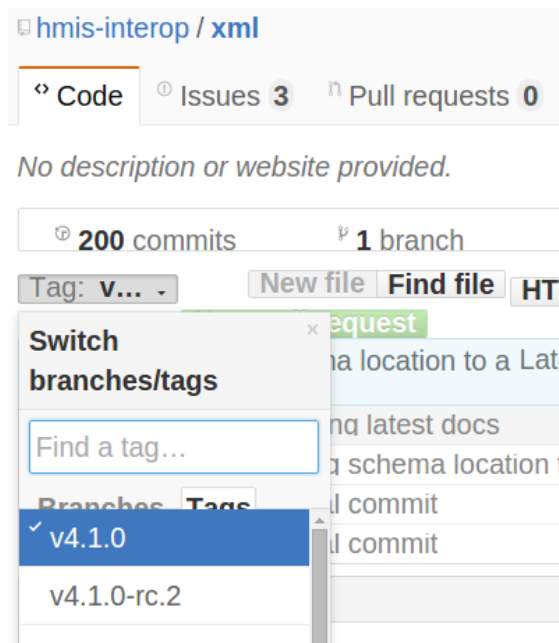
Change Process

To request changes to the HUD HMIS XML Schema, there are multiple ways to register 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 hmisdx@gmail.com (temporary address) to discuss the 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.



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 hmisdx@gmail.com (temporary email address). To log a detailed change directly, an [issue tracker](#) is available.

Overview of Changes

Change List

- Addition SourceType
- Add Rule for Source Type if SourceType = 1
- Add Rule for SourceName if SourceType != 1
- Revise picklist for 3.6.1 Gender to remove "Other" and add substitution "Doesn't identify as male, female, or transgender"
- Remove 3.6.A OtherGender
- Change the numbering to 3.917A.1 for ResidencePrior
- Remove OtherResidencePrior
- Change the numbering to 3.917.2 for ResidencePriorLengthOfStay and revise the picklist to match
- Add LOSUnderThreshold 3.917.2A & 2B
- Change minOccurs for 3.8 DisablingCondition
- Renumber 3.17.1 EntryFromStreetESSH to 3.917.2C and rename to PreviousStreetESSH and change picklist to a three value list from a five value list
- Renumber 3.17.A to 3.917.3
- Renumber 3.17.2 to 3.917.4 and remove one picklist value
- Renumber 3.17.B to 3.917.5
- Change 4.16.A PathReferral picklist #3 to "Primary health/dental care"
- Renumber 4.17.A to 4.17.1 and remove restriction "Null unless InPermanentHousing = 1"
- Add 4.48.1 UrgentReferral
- Add new 4.48.1 "NoPointsYes" picklist which is used by many of the new SSVF elements
- Add 4.48.2 TimeToHousingLoss and picklist
- Add 4.48.3 ZeroIncome
- Add 4.48.4 AnnualPercentAMI and picklist
- Add 4.48.5 FinancialChange
- Add 4.48.6 HouseholdChange
- Add 4.48.7 EvictionHistory and picklist
- Add 4.48.8 SubsidyAtRisk
- Add 4.48.9 LiteralHomelessHistory and picklist
- Add 4.48.10 DisabledHoH
- Add 4.48.11 CriminalRecord
- Add 4.48.12 SexOffender
- Add 4.48.13 DependentUnder6
- Add 4.48.14 SingleParent
- Add 4.48.15 HH5Plus
- Add 4.48.16 IraqAfghanistan
- Add 4.48.17 FemVet
- Renumber 4.44.1 to 4.48.20
- Add 4.48.21 ThresholdScore
- Add 4.49.1 ERVisits and picklist
- Add 4.49.2 JailNights
- Add 4.49.3 HospitalNights
- Change picklist for 4.47.3 ViralLoadAvailable from a standard five value response and check assertion
- Add HouseholdID to EnrollmentCoC to simplify linking CoC information to household members
- Add 4.4.11 IndianHealthServices - also, collapsed noMedicalAssistanceReason, noADAPReason, noHealthInsuranceReason all into reasonNotInsuredOrAssisted picklist
- Add 4.4.11.A dependent NoIndianHealthServicesReason, also added assertion logic to test if 4.4.11 = 0 first

- Add 4.4.12 OtherInsurance
- Add 4.4.12.A OtherInsuranceIdentify
- Change picklists for 4.24.1
- Change picklists for 4.14.A - added all the lists in the documentation, and for clarity, added OtherTypeProvided assertion rule, not enforcing SubTypeProvided assertion (too complex for now, but will enforce when it can be non-null)
- Change picklist 4.16.A
- InPermanentHousing removed in v5.
- Change 4.39 picklist for NoHIVAIDSAssistanceReason
- Folding ContactLocation into ServiceFAReferral RecordType= 12 code list.
- Conditional validation of income sources and insurance
- WorstHousingSituation class renamed to EntryRHSP
- Affiliation broken out as a new class
- Collapse the bedinventory type into the Inventory type per the 2016 Logical Model change
- Add ProjectID to Inventory
- Add ProjectID to CoC type and enforced cardinality
- Update Site's cardinality with CoC
- Rename Services back to ServiceFAReferral
- Add ConnectionWithSOAR to IncomeAndSources
- Add new RecordTypes for Voucher Tracking and related TypeProvided
- Add HUDVASHExitInformation
- ConnectionWithSOAR now its own separate type, and no longer part of IncomeAndSources

Bugs Fixes

- Rename SexualOrientationGenderIdentityYouth to SexualOrientationGenderIDYouth to match CSV (but not Logical Model, which should also change). Also a 4.1/2015 bug.