# HUD HMIS XML
# Version 2.8
# Cumulative Package Overview

Version 2.8: April 2008
Version 2.7: May 2006
First Public Version (2.6): January 2005

*Revisions for* Version 2.8
Prepared for Abt Associates Inc. by
Eric Jahn, Alexandria Consulting LLC

*Original Title:*
*"HMIS Data Integration Cumulative Package Overview"*
*Original Authors:*
Oscar Gutierrez, Ph.D., Change and Strategy Solutions
Brian Sokol, Abt Associates Inc.

U.S. Department of Housing and Urban Development
Office of Community Planning and Development

# HUD HMIS XML CUMULATIVE PACKAGE OVERVIEW

## Preface

**This document is intended for HMIS technology project managers and implementing technicians seeking guidance on the purpose and intended use of the HUD XML Schema Definition (XSD) for HMIS data, version 2.8.** It is not a policy document; much of its language is technical and assumes a basic understanding of the W3C XSD specification. For HMIS implementers seeking a simpler method of transmitting HMIS data, see the HUD Comma-Separated Values (CSV) for HMIS data, version 2.7.[1]

In September 2003, the Center for Social Policy at UMass Boston and HUD released a white paper "HMIS Integration Strategies and Solutions."[2] The paper outlined issues surrounding data integration for a non-technical audience. This document is intended for a more technical audience.

## Section 1: Introduction

Continuums of Care across that nation are struggling to gather more comprehensive data on homeless persons within and across Continuum boundaries. HMIS implementations are therefore struggling with more effective data integration. Within a Continuum of Care, homeless services data are often maintained outside the HMIS in separate databases maintained by individual agencies. Multiple HMIS software solutions may also be used within a single reporting area or Continuum. Both scenarios require data integration to overcome the administrative and technical implementation barriers to comprehensive homelessness reporting.

Before starting this project, HUD's contractors surveyed key stakeholders including software providers and several communities experiencing data integration challenges. Respondents indicated that the most pressing need regarding data integration was creation of an XML[3] Schema Definition (XSD[4]) for HMIS data. Therefore, the current materials define a technical format to represent the data elements required by the *HUD HMIS Data and Technical Standards Final Notice* (hereafter referred to as the *HUD HMIS Notice*[5]). The goal of the format is to provide a single, software application neutral methodology to achieve standardized data integration between diverse systems.

To facilitate operational deletes and updates of small subsets of HMIS data, a Web Services Application Programming Interface (API), or similar technology, is needed. A proposed standard set of instructions for calling such a web service is included with this

---

1 Available at http://www.hmis.info/Default.aspx?classicasp=resources.asp%%resource_id=822

2 http://www.hud.gov/offices/cpd/homeless/hmis/assistance/dataintegration.pdf

3 XML stands for eXtensible Markup Language, an overview of XML is available at: http://en.wikipedia.org/wiki/XML

4 http://www.w3.org/XML/Schema

5 Department of Housing and Urban Development, Homeless Management Information Systems Data and Technical Standards, Final Notice, 69 Fed. Reg. 45889-45934 (July 2004). http://www.hud.gov/offices/cpd/homeless/rulesandregs/fr4848-n-02.pdf

update of the schema, so that vendors and communities may test its suitability for defining operational data transactions.

**Contents of Package**

The current package consists of multiple parts:

- This document, which includes:
  - A rationale for the schema, including an overview of the process and a description and 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 set of two XML Schema Definition (XSD) documents. The main document is the HUD HMIS XSD v. 2.8. The other is the referenced AIRS XSD v. 3.0 draft 5, modified with the addition of a target namespace and the removal of <xs:any/> tags, so the schema may be imported.
- A sample, valid XML document with fictitious data.
- An example extension schema of the HUD HMIS 2.8 XSD, adding an additional element.
- A sample valid XML instance document for the extended schema.
- A proposal for a Web Services Description Language (WSDL[6]) standard HMIS web service description.
- A set of SOAP authentication codes for the web service. See Appendix 1.8.

In addition, online documentation for version 2.8, with graphical representation, is available at: http://www.hmis.info/schema/2_8/docs/HUD_HMIS_2_8.xsd.html

## Section 2: Data Integration Rationale

### I. Process

The XML format is better suited than flat files, e.g. Comma-separated Values[7] (CSV) files, for conveying hierarchical data. The schema can enforce a specific structure to XML data, while there is no accepted standard for enforcing a similar structure for flat files. Thus, the primary product of this effort is an XML Schema Definition (XSD) document.

[6] http://en.wikipedia.org/wiki/Web_Services_Description_Language
[7] http://en.wikipedia.org/wiki/Comma-separated_values

Three concerns guided the original XSD development:

1) Modeling the data precisely as expressed by the *HUD HMIS Notice*. Largely, limited the scope to those data required by the *HUD HMIS Notice,* and left out other data that might be collected on a community level.
2) Accounting for the mechanics of data integration itself within the model. For example, data elements were required to track the originating database of particular records and the date when the data were collected, in order to properly synchronize the data.
3) Creating a clear and simple schema that could be used and understood by all HMIS developers and local database engineers and consultants who may be employed to convert data to the HUD HMIS standard.

Viewing the *HUD HMIS Notice* as the underlying requirements document, the following needs added layers of complexity:

- Certain elements are collected generally once for a person, other elements are collected once per program (site service) enrollment (*HUD HMIS Notice,* Sec. 2.0). A third set of elements are historical data collected multiple times within a single program (site service) enrollment (*HUD HMIS Notice,* Sec. 5.16), for example during site service entry and at exit.
- In several instances, the *HUD HMIS Notice* supplies a list of response codes and mandates that more than one answer must be allowed.
  - Examples of this are "other names" (see chart in *HUD HMIS Notice,* Sec. 2.1), race, educational degrees, income, and school barriers. These elements are relationally modeled as child tables rather than as yes/no fields for each element, which would not have allowed the inclusion of the standard codes HUD requires.[8] In the XSD they are unbounded child elements, either simple or complex as appropriate.
- Since HUD has mandated particular numeric codes for all standard values, the model must enable the transmission of numeric codes, rather than the descriptive values.
- HUD requires that whenever the "other" response category is allowed, a separate text field for a description must be created (HUD HMIS Notice, 5.1.3).

Version 2.8 of the HUD HMIS XSD attempts to incorporate suggestions raised by those who have implemented previous versions of the schema. It also attempts to takes advantage of two of the more powerful features of the XML Schema Definition specification[9]: namespaces and reuse of top level, extensible types. A listing of the specific changes made in Version 2.8 are described in Appendix 1.

[8] In relational modeling, a "child table" refers to a table that can have multiple records compared to a "parent" table, which would have only one record. Thus, if one person can have multiple races, then the person table is the parent table, and the table of the person's races is the child. In XML, when one element appears inside another element, the container element is the parent.

[9] http://www.w3.org/TR/xmlschema-0/

**Note:** This document does not attempt to describe the meaning of data elements that are already included in the *HUD HMIS Notice* or to justify, refine, or clarify any of the data definitions given there. Instead, this document describes the decisions made in modeling the data and highlights data elements that were not in the *HUD HMIS Notice* but were included in the Schema for technical reasons or out of anticipation of need.

## II. Overview of Schema Elements

For readers who are already familiar with XML, the following section describes some aspects of the HUD HMIS XSD. Otherwise, there are many good introductions[10] to XML and XML Schema on the web, which should b**e** read before continuing this document.

**Very few data elements are mandatory**. The mandatory elements are limited to those data that are logically necessary in order to produce meaningful information. The mandatory elements should not be confused with the Universal data elements mandated by HUD. Site services (programs) are mandated to collect Universal data elements, but the entire data set is not invalid if one person's record is missing one data element.

Almost all elements are based on top level types. These top level types are often reused multiple times within this schema, and can be imported into a new schema for extension. Some types are imported from the AIRS Schema[11], so the namespaces “hmis” and “airs” and “xsd” are used to keep elements' origins clearly denoted.

A simplified view of the hierarchy of major complex elements in an XML instance document is shown in Figure 1, followed by a description and rationale for these elements. In the description, each Roman numeral (I, II, III) section represents a cluster of elements grouped for logical reasons, and Arabic numbered (1, 2, 3) sections represent distinct complex element sequences.

**Section 3: XML Instance Element Documentation.**

## I. Integration Structural Elements

These elements collect information specific to the data integration environment, enabling synchronization and the tracking of data to its source, which may be needed for resolving errors or other data validation or quality issues.

### 1. SourceDatabase

*Parent Element:* None, it is the “dataroot” element

[10] http://en.wikipedia.org/wiki/XML_Schema and http://www.w3.org/XML/Schema are two useful sites to begin learning XML Schema.

[11] AIRS is the Alliance of Information and Referral Services, “AIRS is the professional association for over 1,200 community Information and Referral (I&R) providers, primarily in the United States and Canada” http://airs.org/

This element records information about each data source that is participating in the data integration environment. The DatabaseID should be initially assigned by the target database, i.e., the database integrating the data, and it should be unique across the implementation. DatabaseName is simply a string intended to give the DatabaseID a familiar name. The remaining elements before Export hold contact information for the individual directly responsible for the sending database, i.e. a database administrator or the IT department manager.

### 2. Export

*Parent Element:* SourceDatabase
Over time, each database will produce multiple exports. This element records information about the beginning and end of the period for which data were extracted and the actual date of the export. As an alternative, the Export element could be omitted, moving its functionality to the SOAP envelope, as specified in the proposed WSDL standard. For more information see "Proposed WSDL Service Description" in Appendix 1: New Features and Modifications in HUD HMIS XSD 2.8.

## II. Site Service

### 3. SiteService (formerly Program)

*Parent Element:* SourceDatabase
This element captures information about services occurring at a particular site. The site service is defined within the Alliance of Information and Referral Services' (AIRS) XML Schema,[12] and is analogous to "program" as described in the *HUD HMIS Notice*. Site service is used in place of program, since site service is more precisely defined than program, which has acquired numerous divergent definitions. Also, AIRS and HUD HMIS XML data are utilized in common scenarios, and this harmonization of names reduces confusion. The HUD HMIS site service simply extends the AIRS Schema v. 3.0's version of site service so that there is interchangeability between HUD HMIS and AIRS XML data elements. Included within the HUD HMIS XSD's site service are FIPS code, facility code, CoC code, and site service type code. Though seemingly not required by HUD in the *HUD HMIS Notice*, the SiteService element also captures information on target populations and whether the site service serves individuals or families. This information is useful for data analysis and reporting. For example, the Annual Homeless Assessment Report (AHAR) asks for data on families and individuals separately.

The SiteService element also includes a SiteID element, which can be used to designate the physical location at which a SiteService exists. In AIRS terminology, an agency may possess many Sites which in turn may possess many Services, and the SiteService represents the intersection of the Site and Service. Use the AIRS Schema in isolation to represent complex multi-layered parent agencies, sites and site service relationships as AIRS XML.

[12] See http://tinyurl.com/ywsypz for the AIRS Wiki

Site service data will not necessarily need to be sent in every periodic upload. Rather, they can be sent during an initial transfer and only updated if changes occur.

## III. Static Personal Information (formerly Client)

### 4. Person

*Parent Element:* SourceDatabase
This element includes identifying information about an individual and any other personal data that will very rarely change over time or between collecting agencies. As such, only minimal data is associated directly with the Person entity.

All of the identifying elements under Person may occur only once, except Race and OtherNames. The Race element can occur multiple times allowing for multiple races for each person. OtherNames is a complex type containing the various parts of the person's other names. Multiple sets of OtherNames can also occur.

The PersonID is populated from the source database and uniquely identifies a person in the system. The DatabaseID concatenated with the PersonID should avoid PersonID collisions when merging multiple databases. The merged database should implement algorithms to unduplicate across databases using the personal information within the Person element.

To make the schema more flexible for purposes of implementing local policies for de-identification and unduplication, the acceptable formats for the Social Security Number and the child elements of Person allow for a choice of either plain text or hashed values.

```
<xsd:element name="LegalFirstName" type="hmis:hashingChoice" minOccurs="0"</xsd:element>

...

<xsd:complexType name="hashingChoice">
        <xsd:choice>
                <xsd:element name="Unhashed" type="hmis:string50" minOccurs="0"/>
                <xsd:element name="Hashed" type="hmis:string" minOccurs="0"/>
        </xsd:choice>
</xsd:complexType>
```

In the XML instance documents, the various elements contributing to the PersonID hash result may or may not actually have data in them. The reason for this is to provide implementers flexibility in how they want to implement (or not implement) hashing security and unduplication algorithms. One has the choice of:

- Not transmitting personal identifying information for client confidentiality and simply transmitting the hashed PersonID for unduplication

*Example:*

```
<hmis:PersonID>
        <hmis:Hashed hmis:dateCollected="2004-08-01T00:00:00">DCF017Y0</hmis:Hashed>
```

```
</hmis:PersonID>
...
<hmis:LegalFirstName hmis:dateCollected="2004-08-01T00:00:00"/>
```

**Note: omitting the LegalFirstName element entirely is a more efficient transmission format for the above example.*

- Transmitting hashed personal identifying information and also transmitting the hashed PersonIDStr for unduplication

*Example:*

```
<hmis:PersonID>
        <hmis:Hashed hmis:dateCollected="2004-08-01T00:00:00">DCF017Y0</hmis:Hashed>
</hmis:PersonID>
...
<hmis:LegalFirstName>
        <hmis:Hashed hmis:dateCollected="2004-08-01T00:00:00">A2TD87YU</hmis:Hashed>
</hmis:LegalFirstName>
```

- Transmitting clear text personal information and also transmitting the hashed PersonID for unduplication

*Example:*

```
<hmis:PersonID>
        <hmis:Hashed hmis:dateCollected="2004-08-01T00:00:00">DCF017Y0</hmis:Hashed>
</hmis:PersonID>
...
<hmis:LegalFirstName>
        <hmis:Unhashed hmis:dateCollected="2004-0801T00:00:00">Washington-</hmis:Unhashed>
</hmis:LegalFirstName>
```

One could also present clear text PersonIDs as integers or strings with the combinations above. Regardless of the method, the XML instance documents clearly indicate which is occurring.

### 5. OtherNames

*Parent Element: Person*
HUD requires that agencies ask whether persons have received services using any other names. This element is not required for every person but can occur multiple times to hold many additional names.

## IV. Site Service Participation

Certain data required by the *Notice* are specific to a person's participation in a particular program or "site service". These data must be collected once for each person's enrollment in a site service. The data will be collected multiple times if the Person enters the site service more than once. These elements are:

| Element Name | Parent Element |
|---|---|
| VeteranStatus (Universal Element #2.6) | SiteServiceParticipation |
| Disabling Condition (Universal Element #2.7) | SiteServiceParticipation |

| | |
|---|---|
| PriorResidence (Universal Element #2.8) | SiteServiceParticipation |
| IsLastPermanentZIP (Universal Element #2.9) | PersonAddress |
| StartDate (Universal Element #2.10) | ParticipationDates |
| EndDate (Universal Element #2.11) | ParticipationDates |

IsLastPermanentZIP is located within PersonHistorical since it is also related to the PersonAddress complex type. PersonAddress can be collected more often than at SiteServiceParticipation.

### 6. SiteServiceParticipation (formerly ProgramParticipation)

*Parent Element: Person*[13]
SiteServiceParticipation relates a person to a site service and tracks data that are specific to a person's enrollment in a site service.

A Person record may have multiple site service enrollments for different site services, or for the same site service with different dates. SiteServiceParticipation should not be confused with the ServiceEvent element. The former references overall enrollment in a site service, the latter references particular events. Thus, if a site service entails multiple counseling sessions, say one per week over ten weeks, there would be one SiteServiceParticipation record and ten ServiceEvent records.

HouseholdID is included as an element within SiteServiceParticipation to flag an entire household, of which the enclosing Person record is a member, as having participated as well in the same SiteService. The *HUD HMIS Notice* (Section 2.14) states that "a household is a group of persons who together apply for homeless assistance services." Thus, the household membership is not static for a particular person, but is based on a SiteServiceParticipation. Multiple persons in the same household should have the same household identification number. Household identifiers should be generated by the source database. A combination of the site service identifier and the locally generated HouseholdID will be unique across the implementation.

## V. Person Historical Information

### 7. PersonHistorical (formerly ClientHistorical)

*Parent Element:* SiteServiceParticipation or Person
The bulk of the Program-specific data elements collected on each person are captured within the PersonHistorical element and its subelements. Each person can have zero or many PersonHistorical records within each site service Participation.

The PersonHistorical element includes many simple subelements that are collected for all persons, as well as many that are collected only for adults or only for children.

[13] Included by reference to avoid excessive nesting

Three elements—NonCashSourceCode, DegreeCode (for educational degrees), and BarrierCode (for school barriers)—are unbounded elements, allowing for multiple responses. For each of these, the structure allows for one free text element in case the "other" code is used. Thus, for example, the person may have multiple non-cash benefits, including one "other". PersonHistorical also contains two complex subelements as listed below.

PersonHistorical can be transmitted independently of SiteServiceParticipation, and placed directly under a Person element.

### 8. Veteran

*Parent Element:* PersonHistorical
A person may have many Veteran elements or none at all. Veteran has a zero-to-many relationship to the PersonHistorical element. More than one Veteran element are allowed so that multiple service areas and military branches can be captured at any given time. If a person served in two war zones in two different branches of the military, these data would be grouped together inside separate Veteran tags.

Note that the complex Veteran element described here is meant to collect details of military service and is distinct from the simple VeteranStatus element, which is a Yes/No element contained directly under SiteServiceParticipation. This structure reflects the requirements set forth in the *HUD HMIS Notice.*

### 9. IncomeAndSources

*Parent Element:* PersonHistorical
IncomeAndSources is a complex element containing three simple elements. Multiple income sources and amounts can be captured at each point in time. An additional field is grouped with the element to allow for free text description of the income if the code for "other" is used. Following the *HUD HMIS Notice,* the income amounts should be uniformly represented as monthly amounts.

## VI. Needs

### 10. Need

*Parent Element:* SiteServiceParticipation or Person
Needs are not mentioned in the *HUD HMIS Notice*, however fulfillment of personal needs are central to most HMIS software architectures, as well as CoC board reporting. Need elements track AIRS services/service categories a person requires, as opposed to any actual services received. Need elements can be placed either within a SiteServiceParticipation element or outside of a SiteServiceParticipation directly under a Person element, when the need isn't related to a specific site service. A Need element contains many subelements, including NeedID as an index, ServiceEventID to tag specific service events as pursuant to need fulfillment, NeedAIRSCode to hold the AIRS

Taxonomy Code of the need, NeedStatus to indicate fulfillment state of the need, and SiteServiceID to credit a specific site service with registering the need.

### 11. ServiceEvent

*Parent Element:* SiteServiceParticipation
The ServiceEvent element describes particular services actually rendered for a particular person. This entity stores data on the date and type of service, as described by the *HUD HMIS Notice.* An additional field is grouped with the element to allow for free text description of the service if the code for "other" is used.

- To increase the flexibility and usefulness of ServiceEvent data certain elements are included that are not described in the *HUD HMIS Notice.* These include:
- <ServicePeriod>...<EndDate>, which captures the end date of the service.
- ServiceUnit, which indicates whether the ServiceEvent was delivered to an individual or the entire household. In the latter case, the ServiceEvent record need only be included under the head of household. A family apartment unit is an example of a service delivered to an entire household.
- QuantityOfService, which can be used to indicate, for example, dollar amounts if the service is rental disbursements.
- QuantityOfServiceMeasure had been added in version 2.7 to describe the unit of measure used for the QuantityofService element.
- AIRSCode, this string can be used to track the standard taxonomic code from the Alliance or Information and Referral Services (AIRS), e.g., "BH-180" if the service is Emergency Shelter.
- IsReferral, added in Version 2,7, allows a ServiceEvent to be flagged as a referral as opposed to a directly provided service.

## VII. Release Of Information

### 12. ReleaseOfInformation

*Parent Element:* Person
The Release of Information complex element contains a record of a person's consent to share their personal information for a particular SiteService during a defined time frame. ReleaseOfInformation includes subelements ReleaseOfInformationID to index the releases, SiteServiceID to relate the release to a specific site service, and EffectivePeriod to limit the release to a specific time frame.

## VIII. Household

### 13. Household

*Parent Element:* SourceDatabase

This element allows household groupings of persons to be conveyed. Household can contain multiple Member elements, each describing their ID and relationship to the head of household. In the household element, one of the members can have their ID designated as HeadOfHouseholdID. The *HUD HMIS Notice* (Section 2.14) states that "a household is a group of persons who together apply for homeless assistance services." Thus, the household is not static for a particular person, but is based on a site service enrollment. A person can be a member of multiple households. Multiple persons in the same household should have the same household identification number. Household identifiers should be generated by the source database. A combination of the site service identifier and the locally generated HouseholdID will be unique across the implementation.

## IX. Types

With the release of version 2.8, top level types, both simple and complex, exist for every element declared in the schema. This allows any type, including the root element, SourceDatabase, to be imported and referenced or extended within another schema. This enables flexible, validated customization of the HUD HMIS 2.8 Schema, unlike the "Custom" tags present in version 2.7 of the Schema. An example schema demonstrating the new syntax for extending the HUD HMIS 2.8 is included in the data integration package. An example XML instance document which validates against this extended schema is also included in the package.

## X. Lookup Values/Enumerations

These elements declare the acceptable values for the data elements and can be based on patterns or enumerations.

In cases where the *HUD HMIS Notice* defines a specific list of acceptable codes with their description, the schema uses the codes as the acceptable values. The interpretations of the values are included in the XSD file within Documentation tags. This strategy has disadvantages in that it reduces the human readability of a given XML file. That is, an individual file would require the XSD or additional information to decode the XML element values. However, it is clearly in line with HUD's intention in the *HUD HMIS Notice* and it also relieves programmers of the burden of having to convert codes to values and back again to codes.

Multiple data elements might reference the same lookup enumeration if they use the same code breakdown. For example, multiple fields accept the values 1, 2, 8 or 9 even if the meanings of these values differ. (The use of the non-sequential 8 and 9 values to indicate "Don't Know " and "Refused" is mandated by the *HUD HMIS Notice* and has the advantage of consistency across elements, even though it is somewhat non-intuitive within each element.)

Where enumerations are linked to multiple descriptions, they are named after the number of meaningful values. For example, "twoval" refers to an enumeration with only two

acceptable values; "twovalplus" refers to elements with two meaningful values plus allowable values for "don't know" and "refused."

## Section 4: Beyond the Current Package

This data integration package is only the first step in achieving full data integration. The current specification assumes that participating stakeholders agree to use the standard XML description and will have software convert their stored data to the standard XML format. Many more steps are needed in order to actually complete the integration process. Technical steps include implementing a process for validating the data, transferring the data via a standard communication protocol (commonly SOAP or REST messaging), creating a database to act as a central repository, and devising a synchronization method. In addition to the technical steps, a number of decisions need to be made and responsibilities divided between the central agency and the participating parties.

### I. Validation

Before sending the XML, the contributing data source should validate the files. The following list presents the most commonly used XML parsers to date. Most of the tools listed here can test XML documents for being not only well formed (i.e. conform to the basic syntax of XML) but also valid (they conform to a particular schema or DTD).

| Validating Parser | URL |
|---|---|
| CFX XML Parser (Coldfusion) | http://www.zrinity.com/xml/xmlparser |
| Crimson | http://xml.apache.org/crimson |
| JAXP | http://java.sun.com/webservices/jaxp |
| libxml | http://xmlsoft.org/ |
| lxml | http://codespeak.net/lxml |
| MSXML | http://msdn2.microsoft.com/en-us/xml |
| Oracle | http://www.oracle.com/technology/tech/xml |
| Xerces | http://xerces.apache.org/xerces2-j |
| XSV (W3.org official reference implementation for validating the XML Schema) | http://www.ltg.ed.ac.uk/~ht/xsv-status.html<br>convenient web form:<br>http://www.w3.org/2001/03/webdata/xsv |

In addition to validating the XML against the HUD HMIS XML Schema, the data should also be scrubbed to ensure that the content of the data is reasonable. Additional checks might be appropriate to ensure, for example, that birth dates are earlier than the current date, or that only women are marked as pregnant. This is beyond the level of validation handled by the parsers and schema.

### II. Processes for Data Transfer

There are varying levels of technological sophistication that can be implemented for transferring the data.

## 2. Simple: Unidirectional Batched Data Uploads

The least technically sophisticated integration approach would involve minimal automation and rely heavily on people doing the work. Consequently, data would be sent infrequently, such as quarterly or yearly. The process would be one-way. Participating agencies will not receive data back from the central repository.

Database developers of each aggregate database will develop a process for exporting the data to the XML file and provide an interface for users of the system to easily export the data for a certain date range. Many database tools have functionality to export data as XML. In addition, various tools can assist developers with this conversion process. The exported data file is uploaded via secure FTP or SSH to an Internet site managed by the central repository. In very-low level implementations the file can even be emailed, as long as the email or the attached file is encrypted during delivery transport, with at least 128-bit encryption.

## 2. Complex: Bidirectional Asynchronous[14] Messaging

A fully integrated environment will be one that satisfies the three following characteristics. First, it allows for data to be transmitted to and from source and destination systems. Second, it is an environment where multiple human services domains such as mental health, substance abuse, and health care - in addition to homelessness - contribute data to an aggregate database. Third, it is a technically flexible environment where heterogeneous data management applications that operate under diverse messaging protocols can asynchronously push and pull data sets according to standard security and authorization mechanisms.

For this environment to be put in effect, developers at the local level must assess the most appropriate tools to implement asynchronous data transmission; develop both push and pull data extraction utilities; and integrate the array of necessary data messaging services that are more applicable to the locality.

In this environment, data transmission is bidirectional; in other words, a participating system contributes with data to an aggregate database, but also has the ability to extract up-to-date data sets from the aggregate database for analysis or consolidation purposes. Bi-directional does not necessarily mean that communication between participating and integrating systems is synchronous (i.e. the ability to handle two-way communication on a real time basis). However, the ultimate data integration environment is a flexible, asynchronous, bidirectional data transmission system.

Messages should implement the basic ACID[15] record functions of “add, change, inquire, and delete”, and only necessary portions of valid XML, such as a single ServiceEvent, may be present in a single message.

---

[14] Asynchronous means the sending server need not block/wait for the receiving server to respond before continuing on with the next task.

[15] http://en.wikipedia.org/wiki/CRUD_(acronym)

Commonly used messaging protocols include SOAP and REST. These protocols are often referred to as web services, as they overlay the internet as the underlaying physical transport. SOAP and REST attempt to accomplish the same means and they are competing messaging methods. A draft SOAP Application Programming Interface is included with this 2.8 HMIS Schema release, so that implementers can test and improve the API. A REST interface should also be contributed and tested by implementers, as REST may become the predominant web services messaging protocol.

## III. Converting from XML to Databases

The administrator of the central repository creates the aggregate database to store the data sent by different data sources. In general, there are two approaches to storing XML in databases: data-centric and document-centric. In the document-centric model, the database stores each XML file as a complete document or in a manner that is easily suited to retrieving the data again as a separate XML file. The data-centric model stores the data within the XML document rather than the XML document itself. The data-centric model is most appropriate for HMIS purposes since the primary data source is generally a relational database.

The process of transferring data to a relational database is called “shredding” XML into tables. Most major relational database systems have built-in functionality to handle this process, especially if the database schema maps closely to the XML schema. A closely mapped database would consist of separate tables for every complex element as well as every element that may occur multiple times. The columns in each table will consist of all of the simple elements contained with the complex elements. The entity relationship diagram included with this document is an example of a database schema that maps to the HMIS XML schema.

Most major databases are “XML-enabled” which means that they have some ability to export data to XML and import data from received XML. But, different relational database systems handle import of XML in very different ways. (In addition, a wide array of middleware products is available to assist with the process of mapping between XML schemas and databases. A good overview of the relationship between XML and databases as well as lists of databases and middleware that support conversion between XML and databases can be found at http://www.rpbourret.com/xml/XMLAndDatabases.htm .

The process of importing the data may also incorporate the matching of client records immediately. In that case the import script would include the process of searching the current database to determine whether the person is already present. If so, the person record would be updated rather than inserted, and the database would be unduplicated at all times. Alternatively, it is possible to simply insert all records into the aggregate database when they are initially received, and at a later point, match the records. This decision will impact how the aggregate database is designed.

## IV. Documentation of Responsibilities and Decisions

Project documentation should be created to help guide administrators of participating data sources. The following is an outline of the responsibilities that need to be clarified and decisions that to be made:

*a. Converting the data*

- ✓ Whose job will it be to convert the data in source databases to the data standard?
- ✓ How much support will be provided by the staff of the central database?

*b. Ensuring cross-participant consistency*

- ✓ How will questions and answers about the meaning of the fields be managed?
- ✓ Will the implementation make efforts to ensure that all participating source databases understand the data standard in the same way?

*c. What messaging protocol will be used (SOAP, REST, other)?*

*d. Hashing the data*

- ✓ The standard includes the option for certain values to be hashed within the file. Which site services will be allowed to send hashed values?

*e. Cleaning and validating the data*

- ✓ Who is responsible for cleaning the data and removing invalid or incorrect information?
- ✓ How clean must the data be before being sent to the central agency?
- ✓ Should the data be converted to a certain format (e.g., upper case) before being sent?
- ✓ Beyond the technical requirements of the standard, what, if any validation rules should be applied (e.g., acceptable dates for date of birth, incomes within reasonable ranges, pregnancy only for women)? Should this validation occur prior to sending the data?

*f. Unduplicating the original data.*

- ✓ Should the data sent to the central database be unduplicated before being sent, so that the central database can treat all distinct person records sent from a particular agency as unique. Or should the data source extract all of the records and leave unduplication to the central repository.

*g. Frequency*

- ✓ How often should the data be sent?

*h. Filtering the appropriate data for a period*

- ✓ What records should be sent?
- ✓ If batch data are sent every month, should data be sent on every person that was served that month? Or only those who entered, exited, or had their records updated that month?
- ✓ Should a complete data dump be sent every time, leaving the central repository to filter based on dates in the system?

*i. Sending the data*

- ✓ What are the protocols for sending the data?
- ✓ What methods are available

*j. Handling data conflicts and synchronization*

- ✓ How should data conflicts be resolved (for data that are not longitudinal)?

*k. Securing the database*

- ✓ What information security protocols will be in place at the central database?
- ✓ Under what circumstances and in what manner will the data be released?

These decisions can be made by the individuals overseeing the integration effort, though some might require community input. Answers to the questions can be written in a Standards Manual or Frequently Asked Question (FAQ) format. The outline above can be used as the starting point for this documentation. An overview of the data integration process responsibilities can be found at:

- http://hmis.info/ClassicAsp/documents/8 Simple Rules - Data Integration Institute.pdf
- http://hmis.info/ClassicAsp/resources.asp?resource_id=860

## V. Recommendations for Continued Standardization

The current project was limited to development of an extensible data schema and a draft proposal for messaging. Additional standardization can help to move integration projects further down the road to success. A voluntary working group could continue to develop such standards and move them forward. Ultimately, such a working group can also be charged with maintaining the schema. In the meantime, individual implementations using this XML Schema will need to establish local protocols, and ongoing feedback on the schema and SOAP API should be submitted to the authors or through the HUD HMIS Technical Assistance process. In addition, the development of both proprietary and non-proprietary software and the sharing of code to support this schema on various platforms are highly encouraged. A section of the HMIS Technical Assistance portal has been made available for these purposes.

## Appendix 1: New Features and Modifications in HUD HMIS XSD 2.8

The 2.8 Schema introduces a number of significant changes and several minor adjustments. These changes are described below. Complete online graphical documentation of the Schema is available at:
http://www.HMIS.info/XMLDocumentation/2.8/HUD_HMIS_2.8.xsd.html

### 1. Need Element

A Need element has been added to the HUD HMIS 2.8 Schema. See Section VI. 10. above for an explanation of its purpose.

### 2. Program Renamed to SiteService

A SiteService in HUD HMIS XML now has the same meaning as a SiteService in AIRS XML, which is a standard for describing service provider information. A SiteService's parent element can only be an AIRS "Site", for which there is also a data element in HUD HMIS XSD v. 2.8. The HUD HMIS 2.8 Schema does extend the AIRS definition of a site service to add some HUD-specific data elements such as the FIPS and COC codes, but all the base AIRS elements are included in the HMIS 2.8 Schema. If more complex agency, site, and Service relationship hierarchies need to be conveyed between systems using XML, AIRS XML could be transmitted in conjunction with HUD HMIS XML.

### 3. ProgramParticipation Renames to SiteServiceParticipation

The former ProgramParticipation has been renamed to SiteServiceParticipation to eliminate ambiguity in the relationship between HUD XML and AIRS XML.

### 4. PersonHistorical Made Independent of SiteServiceParticipation

PersonHistorical has replaced the former ClientHistorical, since the confusingly dual-named "Client" and "Person" elements were both in the prior schema. PersonHistorical now captures any type of dynamic (changing) data about an individual over time and could be extended by an implementer to add custom elements.

### 5. PersonHistorical Elements are Unattached to SiteServiceParticipation

PersonHistorical can optionally be placed within a SiteServiceParticipation. This flexibility allows information about a person to be recorded without requiring an an associated SiteService entry. PersonHistorical may still exist within a SiteServiceParticipation, as before, to associate the data with a SiteService entry/exit. In this situation, the dataCollectionStage attribute within the associated dateStampGroup can capture whether data was collected at SiteService entry, exit, enrollment, or follow-up.

## 6. An Independent Household Element

A household element independent from the Person element is now included in the HUD HMIS 2.8 schema. It can contain multiple Member elements, each describing their RelationshipToHeadOfHousehold. In the household element, one of the members can have their ID designated as HeadOfHouseholdID.

## 7. Export Element Now Optional

The export element is now minOccurs="0" so that it may be left out of the XML instance document in favor of including export information within the SOAP message envelope. See proposed WSDL service description section below.

## 8. Proposed WSDL Service Description

There has been an increasing need for operational data integration beyond simple data warehousing for reporting purposes. To meet this need, a proposed service definition for interfacing with an HMIS, called a Web Services Description Language (WSDL[16]) is included with the 2.8 HUD HMIS data integration package. While it is just a proposal, HUD recommends HMIS implementations attempt to implement the service calls described in the WSDL. Code needs to implement the operations described in the WSDL. Software is available which generates appropriate service stubs for a given valid WSDL file. In brief, the WSDL describes create, get , update, and delete operations for each of the major complex types described in the schema.

For example the WSDL describes how to request that the target HMIS create a new Need with a Service Event. All of these requests are transmitted in a SOAP envelope which comprises the message to the target HMIS. Within the SOAP envelope request, complete or simply portions of valid HUD HMIS XML are enclosed. The SOAP envelope, along with its payload, contains export transmission details, replacing the functionality of the export section in the schema.

In response to transmitted SOAP messages, authentication codes are returned. These standard codes are enumerated in the self-named file included in the Data Integration Package.

To learn how to fully interpret the instructions in the WSDL file included in the HUD HMIS XML Data Integration Package, follow the links at http://en.wikipedia.org/wiki/Web_Services_Description_Language . The following diagram illustrates the major sections of the WSDL.

[16] For more information on the Web Services Description Language, see http://en.wikipedia.org/wiki/Web_Services_Description_Language

*Figure 2: Diagram of WSDL Structure*

```
<wsdl:definitions name="HUD_HMIS_2.8_Service"...
        <wsdl:types>
Types are the basic building blocks for the rest of the document.  These types are essentially the same as types in XML Schema.
        </wsdl:types>
        <wsdl:message name="PostDocAttachment_In">
Many messages containing multiple parts are defined at this location in the document.  A message is the basic request, specifying an input or output type defined above.
        </wsdl:message>
        <portType name="HMIS_PortType">
A port type bundles together the messages into input message/ouput message groups, describing a full interaction with the server implementing the WSDL described Web Service.
        </portType>
        <wsdl:binding name="HMIS_Binding" type="tns:HMIS_PortType">
Bindings specify the transport (HTTP in the case of this WSDL, but it could have been SMTP or another transport type).
        </wsdl:binding>
        <service name="HMIS_Service">
The service give the binding a URL.
        </service>
</wsdl:definitions>
```

## 10. Addition Of Release Of Information

A release of information section was added so HMIS systems could share client releases between themselves. See complex element description in Section 3: VII. 12.

## 11. IDs Added To All Complex Elements

To help receiving databases index the various data, index identifiers were added to all complex elements in the schema. The specific indexes which now exist are PersonHistoricalID, DatabaseID, ExportID, HouseholdID, NeedID, SiteServiceID, PersonID, SiteServiceParticipationID, ReleaseOfInformationID, and ServiceEventID.

Each of these indexes are of the type “id”. The definition of this type is below, constraining “id” to be either an integer or a 32 character string.

```
<xsd:complexType name="id">
        <xsd:choice>
                <xsd:element name="IDNum" type="hmis:integer"/>
                <xsd:element name="IDStr" type="hmis:string32"/>
        </xsd:choice>
</xsd:complexType>
```

## 12. Schema Target Namespace

In prior versions of the Schema, a targetNamespace had not been declared in the schema declaration. Now, for schema extension and reuse, a targetNamespace is declared to make schema type importation and "includes" possible. This targetNamespace is then associated with the prefix "hmis:":

```
<schema ...
targetNamespace="http://hmis.info/xsd/hud_hmis_2.8.xsd"
xmlns:hmis="http://hmis.info/xsd/hud_hmis_2.8.xsd"
...</schema>
```

## 13. All Elements And Attributes Are Namespace Qualified

All elements and attributes now must be qualified with a namespace to reduce confusion as to the origin of an entity when extending the Schema, and within the Schema. All elements and attributes originating within the 2.8 Schema are qualified with the prefix "hmis:". Here is the code in the Schema declaration that enforces this:

```
elementFormDefault="qualified"
attributeFormDefault="qualified"
```

## 14. The dateStampGroup Attribute Group

For database synchronization purposes, especially in an operational integration scenario, all subelements within a HUD HMIS XML instance document now may possess a set of attributes to establish the timing and intake stage of their creation: dateCollected, dateEffective, and dataCollectionStage. Only dateCollected is required, and dateEffective is only used when the actual data entry date post-dates the actual date the data was collected, or if a case manager wishes to back-date the data to when it became valid. The dataCollectionStage indicates whether the data was captured during site service entry, site service enrollment, exit from the site service, or site service followup. Here are the attribute definitions from the XML Schema v.2.8:

```
<xsd:attribute name="dateCollected" type="xsd:dateTime" use="required"/>
<xsd:attribute name="dateEffective" type="xsd:dateTime" use="optional"/>
<xsd:attribute name="dataCollectionStage" type="hmis:fourValBase" use="optional"/>
```

A resulting XML example utilizing all of the dateStampGroup attributes:

```
<hmis:Destination hmis:dateCollected="2004-08-01T00:00:00" hmis:dateEffective="2004-07-01T00:00:00"
hmis:dataCollectionStage="1">3</hmis:Destination>
```

## Appendix II: Minor Modifications

### 1. Date Ranges

In version 2.7 of the HUD HMIS Schema, multiple elements tracked start and end dates for ProgramParticipation, ExportPeriod, etc.. In 2.8 a generic dateRange type is used for all date ranges, but there are two variants: dateRangeCapped and dateRangeOpen. dateRangeCapped has mandatory start and end dates. dateRangeOpen allows for either one or both of start and end dates to be present, for open ended or uncertain date ranges.

### 2. Export Period Format Change

ExportPeriodBegin and ExportPeriodEnd dates are now folded into ExportPeriod, which consists of a standard dateRange type.

### 3. Entry and Exit Date Format Change

Site service entry and exit dates are now folded into ParticipationDates, which consists of a standard dateRange type.

### 4. Camel-cased Types

All types are now in camel-case. For example, type="twoval" is now type="twoVal". This is an XML Schema convention.

### 5. Removal of the “dataroot” Element

SourceDatabase is the new top level element instead of dataroot.

### 6. ZIP Codes

ZIP Code is now a capitalized acronym.

### 7. DateOfService and DateOfServiceEnd Removed

These two elements have been replaced by ServicePeriod of the type dateRangeOpen which contains optional StartDate and EndDate elements.

### 8. HouseholdID added to SiteService

HouseholdID was added to allow for situations where ServiceUnit =”0” (i.e. where an entire household is receiving a service). The HouseholdID in this context indicates the entire family receives the indicated service.

### 9. Removal of ProgramParentID

ProgramParentID has been removed from the HUD HMIS Schema, as this functionality of describing relationships between programs/site services is more appropriately the

domain of the SiteService element in the AIRS XSD. In other words, use the AIRS XSD to transmit agency/site/service hierarchies between systems.

## 10. Removal of Custom tags.

Custom tags, originally introduced in version 2.7, have been entirely eliminated in favor of extensibility via importation of the HUD HMIS Schema into a new custom third party XML Schema. To accomplish this, types have been prefixed with their namespace, the schema has been given a target namespace, and all types are defined at the top level of the schema so they can be imported and extended. Because of this, customizations can then be validated, which was not the case with the former Custom tags.