Emergency Data Exchange Language (EDXL) Hospital AVailability Exchange (HAVE) Version 2.0

Working Draft 01

01 May 2013

Technical Committee:

[OASIS Emergency Management TC](http://www.oasis-open.org/committees/emergency/)

Chairs:

Darrell O'Donnell ([darrell.odonnell@continuumloop.com](mailto:darrell.odonnell@continuumloop.com)), Individual

Don McGarry ([dmcgarry@mitre.org](mailto:dmcgarry@mitre.org)), [Mitre Corporation](http://www.mitre.org/)

Editors:

Brian Wilkins ([bwilkins@mitre.org](mailto:bwilkins@mitre.org)), [Mitre Corporation](http://www.mitre.org/)

Lizzie DeYoung ([edeyoung@mitre.org](mailto:edeyoung@mitre.org)), [Mitre Corporation](http://www.mitre.org/)

Additional artifacts:

This prose specification is one component of a Work Product which also includes:

* XML schemas: (list file names or directory name)
* Other parts (list titles and/or file names)

Related work:

This specification replaces or supersedes:

* *Emergency Data Exchange Language (EDXL) Hospital AVailability Exchange (HAVE) Version 1.0*. 22 December 2009. OASIS Standard Incorporating Approved Errata. <http://docs.oasis-open.org/emergency/edxl-have/v1.0/errata/edxl-have-v1.0-os-errata-os.html>

This specification is related to:

* *Emergency Data Exchange Language (EDXL) Distribution Element v1.0*, <http://docs.oasis-open.org/emergency/edxl-de/v1.0/EDXL-DE_Spec_v1.0.pdf>
* *Emergency Data Exchange Language (EDXL) Resource Messaging v1.0*, <http://docs.oasis-open.org/emergency/edxl-rm/v1.0/errata/EDXL-RM-v1.0-OS-errata-os.html>
* *Emergency Data Exchange Language Common Types v1.0*, <http://docs.oasis-open.org/emergency/edxl-ct/v1.0/edxl-ct-v1.0.html>
* *Emergency Data Exchange Language Customer Information Quality v1.0*, <http://docs.oasis-open.org/emergency/edxl-ciq/v1.0/edxl-ciq-v1.0.html>

Declared XML namespaces:

* list namespaces declared within this specification

Abstract:

Summary of the technical purpose of the document.

Status:

This [Working Draft](http://www.oasis-open.org/committees/process.php#dWorkingDraft) (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or [approved](http://www.oasis-open.org/committees/process.php#committeeDraft) as a Committee Draft (Committee Specification Draft or a Committee Note Draft). The OASIS document [Approval Process](http://www.oasis-open.org/committees/process.php#standApprovProcess) begins officially with a TC vote to approve a WD as a Committee Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Draft.

Initial URI pattern:

http://docs.oasis-open.org/emergency/edxl-have/v2.0/csd01/edxl-have-v2.0-csd01.doc

(Managed by OASIS TC Administration; please don’t modify.)

Copyright © OASIS Open 2013. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](http://www.oasis-open.org/policies-guidelines/ipr) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

[1 Introduction 4](#_Toc369013763)

[1.1 Purpose 4](#_Toc369013764)

[1.2 History 4](#_Toc369013765)

[1.3 Structure of the EDXL Hospital Availability Exchange Specification 5](#_Toc369013766)

[1.4 Terminology 5](#_Toc369013767)

[1.5 Normative References 5](#_Toc369013768)

[1.6 Non-Normative References 6](#_Toc369013769)

[2 Design Principles & Concepts (non-normative) 7](#_Toc369013770)

[2.1 Requirements for Design 7](#_Toc369013771)

[2.2 Example Usage Scenarios 7](#_Toc369013772)

[2.2.1 Day-to-Day – Dialysis Patient: 7](#_Toc369013773)

[2.2.2 First Responder – Responding with Critical Care 8](#_Toc369013774)

[2.2.3 Mass-Scale Vaccination Clinics 8](#_Toc369013775)

[2.2.4 Disaster Response: 8](#_Toc369013776)

[2.2.5 Exercises 8](#_Toc369013777)

[3 EDXL HAVE 9](#_Toc369013778)

[3.1 HAVE Report Definition (non-normative) 9](#_Toc369013779)

[3.2 Supporting Elements (non-normative) 9](#_Toc369013780)

[3.2.1 Common Types 9](#_Toc369013781)

[3.2.2 Selecting Values from Lists 10](#_Toc369013782)

[3.2.3 ValueListType 11](#_Toc369013783)

[3.2.4 ValueKeyType 12](#_Toc369013784)

[3.2.5 EDXL Extensions (needed?) 12](#_Toc369013785)

[3.3 Element Reference Model (non-normative) 12](#_Toc369013786)

[3.4 Distribution of EDXL-HAVE (non-normative) 13](#_Toc369013787)

[3.5 HAVE Elements 13](#_Toc369013788)

[4 Data Dictionary (normative) 14](#_Toc369013789)

[4.1 HAVE Message 15](#_Toc369013790)

[4.2 Facility 15](#_Toc369013791)

[4.3 Services 16](#_Toc369013792)

[4.4 Resources 16](#_Toc369013793)

[4.5 Operations 16](#_Toc369013794)

[4.6 Staffing 16](#_Toc369013795)

[5 Conformance 17](#_Toc369013796)

[6 18](#_Toc369013798)

[Appendix A. EDXL-HAVE EXAMPLE (NON-NORMATIVE) 19](#_Toc369013799)

[Appendix B. Service Types and Capacities (NON-NORMATIVE) 20](#_Toc369013800)

[Appendix C. Acknowledgments 21](#_Toc369013801)

[Appendix D. Title Text 22](#_Toc369013802)

[D.1 Subsidiary section 22](#_Toc369013803)

[D.1.1 Sub-subsidiary section 22](#_Toc369013804)

[Appendix E. Revision History 23](#_Toc369013805)

# Introduction

## Purpose

The ongoing goal of the Emergency Data eXchange Language (EDXL) project is to facilitate emergency information sharing and data exchange across the local, state, tribal, national and non-governmental organizations of different professions that provide emergency response and management services. EDXL accomplishes this goal by focusing on the standardization of specific messages (messaging interfaces) to facilitate emergency communication and coordination particularly when more than one profession or governmental jurisdiction is involved.

The current roster of published EDXL Standards includes:

* The Common Alerting Protocol v1.2 specification (EDXL-CAP), with various dedicated profiles
* The Distribution Element Specification v2.0 (EDXL-DE)
* The Hospital AVailability Exchange specification v1.0 (EDXL-HAVE)
* The Resource Messaging specification v1.0 (EDXL-RM)
* The Situation Reporting specification v1.0 (EDXL-SitRep)

The primary purpose of EDXL-HAVE is to provide an XML-based reporting format that allows information to be shared about a set (network?) of health facilities including the communication of the status of a health facility, its services, and its resources. These include bed capacity and availability, emergency department status, staffing levels, available service coverage, and the status of a health facilities operations and resources.

The primary audience for EDXL-HAVE is the broad community that interacts with health facilities and it is intended to be used as a tool to automate information flow in and out of the health network. It is not intended to be a tool used for internal administration of health facilities as other standards organizations (e.g. <<NEED HL7 REFERENCE>>).

## History

In a disaster or emergency situation, there is a need for hospitals to be able to communicate with each other, and with other members of the emergency response community. The ability to exchange data in regard to hospitals’ bed availability, status, services, and capacity enables both hospitals and other emergency agencies to respond to emergencies and disaster situations with greater efficiency and speed. In particular, it will allow emergency dispatchers and managers to make sound logistics decisions - where to route victims, which hospitals have the ability to provide the needed service. Many hospitals have expressed the need for, and indeed are currently using, commercial or self-developed information technology that allows them to publish this information to other hospitals in a region, as well as EOCs, 9-1-1 centers, and EMS responders via a Web-based tool.

The Hospital Availability Exchange standard was created to make sharing information about the state of hospitals for day-to-day and crisis use. Initially it was focused purely on hospitals but it has been extended to handle sharing information about the broader health network, including long-term care facilities, urgent care clinics, and temporary aid centres.

HAVE 1.0 was released on 22DEC2009. Since the release of HAVE 1.0 there have been multiple operational uses of HAVE, including after the 2010 Haiti Earthquake. In many of the operational uses there were modified schema used to add services that were not in HAVE 1.0 and to convey other aspects of the data and to handle the sharing of information about non-hospital facilities (e.g. clinics, temporary locations). The use of the HAVE 1.0 standard was encouraging but the shortfalls needed to be addressed. To that end, in 2010 the OASIS EM-TC voted to re-open the HAVE standard with the goal of creating a HAVE 2.0 standard

## Structure of the EDXL Hospital Availability Exchange Specification

The EDXL-HAVE 2.0 standard document structure is defined using successively more detailed or constrained artifacts in the form of textual descriptions, diagrams, figures, tables and Appendices. The EDXL-HAVE XML Schema is provided separately. The overall structure of the EDXL-HAVE report is first represented in an Element Reference Model (ERM). The ERM is the foundation from which individual constraint schemas (individual situation report types) are defined.

The structure of the EDXL-HAVE standard is defined in the following sections:

* Section 2 summarizes the design principles of the standard and shows several usage scenarios;
* Section 3 provides and informal overview of EDXL-HAVE. In particular:
  + Section 3.1 presents an extensive definition of a HAVE report;
  + Section 3.2 describes essential supporting elements in the EDXL Common Types collection, including the use of EDXL Extensions;
  + Section 3.3 presents the Element Reference Model (ERM) which shows the abstract structural relationships of the main components of EDXL-HAVE;
  + Section 3.4 discusses how the distribution requirements for EDXL-HAVE messages may be met through several mechanisms, including EDXL-Distribution Element (DE) and as general data payloads;
  + Section 3.5 presents a summary of the elements that make up a HAVE message.
* Section 4 The Data Dictionary formally defines each element contained in the EDXL-HAVE standard message.
* Section 5 provides conformance information.

These sections together define the message structure, message element definitions, optionality and

cardinality.

## Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

In addition, within this Specification, the keyword “CONDITIONAL” should be interpreted as potentially “REQUIRED” or “OPTIONAL” depending on the surrounding context. The term “REQUIRED” means that empty elements or NULL values are NOT allowed.

## Normative References

[RFC2119] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.

[EDXL-CIQ] <<reference for CIQ>>

[EDXL:

## Non-Normative References

[HL7] Health Level Seven International - <http://www.hl7.org/>

**NOTE: The proper format for citation of technical work produced by an OASIS TC (whether Standards Track or Non-Standards Track) is:**

**[Citation Label]**

Work Product [title](http://docs.oasis-open.org/specGuidelines/ndr/namingDirectives.html#workProductName) (italicized). Approval date (DD Month YYYY). OASIS [Stage](http://docs.oasis-open.org/specGuidelines/ndr/namingDirectives.html#stage) Identifier and [Revision](http://docs.oasis-open.org/specGuidelines/ndr/namingDirectives.html#revision) Number (*e.g.*, OASIS Committee Specification Draft 01). Principal URI ([version-specific URI](http://docs.oasis-open.org/specGuidelines/ndr/namingDirectives.html#this-version), *e.g*., with filename component: somespec-v1.0-csd01.html).

For example:

**[OpenDoc-1.2]** *Open Document Format for Office Applications (OpenDocument) Version 1.2*. 19 January 2011. OASIS Committee Specification Draft 07. <http://docs.oasis-open.org/office/v1.2/csd07/OpenDocument-v1.2-csd07.html>.

**[CAP-1.2]** *Common Alerting Protocol Version 1.2*. 01 July 2010. OASIS Standard. <http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html>.

# Design Principles & Concepts (non-normative)

Below are some of the guiding principles behind the development of EDXL-TEP:

* Support day-to-day and crisis use of the standard.
* Facilitate sharing of information amongst the general public, all levels of government, first nation/tribal, international, and non-governmental organizations.
* Provide a simple information report that allows first responders, emergency managers, community leaders, politicians, and other stakeholders to get a quick glimpse of the state of the health network in a community.
* Provide a non-invasive way for a health facility to keep the communities that they serve abreast of developments that impact their ability to provide care.
* Be respectful of the boundaries of internal health facility information and the information that is relevant externally.
* Separation of EDXL-HAVE reports from being tied to a particular method of delivery.
* Use and reuse of data, content, and models developed by other initiatives that align with EDXL-HAVE.
* Provide a baseline set of services, operations, and resources to allow health facilities to start using HAVE quickly, while allowing for controlled extension where warranted.

## Requirements for Design

The OASIS EM-TC tasked the EDXL-HAVE Sub-committee to review HAVE 1.0 and propose Errate, Minor, and Major versions. The detailed tasking can be reviewed at:

<https://www.oasis-open.org/apps/org/workgroup/emergency-have/download.php/38356/EDXL-HAVE-SC-SOP_DRAFT_dpm.docx>

## Example Usage Scenarios

The following scenarios illustrate how EDXL-HAVE 2.0 can be used in the field.

### Day-to-Day – Dialysis Patient:

On a routine pickup a social worker picks up an elderly patient that needs routine maintenance. Normally the dialysis is performed at the closest facility, but the social worker knows that the small facility’s dialysis unit is not operating due to an equipment failure. A quick query to view the local health facilities presents several within a 20-minute drive, sothe social worker places a call and coordinates with one of the alternate facilities.

### First Responder – Responding with Critical Care

As the result of a multi-unit residential fire, ambulances are dispatched and the Incident Commander indicates that there are 2 critical and 3 serious burn victims. The nearest hospital can only take in 2 burn victims normally, but the current state of the burn unit is not known. By examining the state of the local facilities, officials can coordinate which victims are to be taken to the surrounding health facilities.

### Mass-Scale Vaccination Clinics

Under pandemic conditions a community is implementing a vaccination program with the hospitals, urgent care clinics, private clinics, and temporary clinics providing vaccinations. The public, key officials, and the media can have immediate visibility into the wait times and service availability at each of the vaccination sites. EDXL-HAVE provides the ability to display service availability for each facility, referenced on a map, by colour code and to provide an indication of wait times if they are available.

### Disaster Response:

Following a major earthquake in the developing world, NGOs, various government responders, and local officials (and non-officials) establish temporary health-care facilities to meet the urgent and non-urgent health needs of those injured or killed by the earthquake and ensuing issues. Coordination of multiple dimensions are critical: what services are available, what is the capacity of the facilities, what resources they are missing or can share, where are the facilities located, who are the official points of contacts, what agency is running the facility, what are the hours operation, etc. <<INTENT here is to cover the disaster scenario…>>

As the event unfolds there is a Cholera outbreak due to damaged sanitation. There is a clear need identified to track 2 particular services (e.g. Cholera Vaccination and Cholera Treatment) that were too specific to be part of the default HAVE 2.0 services taxonomy. After a meeting of the coordinating agencies, the data being shared is extended to include Cholera Vaccination and Cholera Treatment services, including the standard metrics (capacity, colour code for status, etc.) << INTENT of this section is to communicate that HAVE 2.0 can adapt to in-field needs >>

# EDXL HAVE

Section 3 of this Standard is ***normative unless otherwise stated***. If any differences are found between any XML schema and its associated model, diagram, table or other artifact or text, then the XML schema shall always take precedence and the other artifact(s) must be changed to match the XML schema.

Note: Please report any such errors to OASIS.

## HAVE Report Definition (non-normative)

The HAVE Report is a single EDXL message that is intended to provide sharing of the services, operations, and capacities of health facilities. Health facilities in HAVE include hospitals, urgent care clinics, temporary facilities, and other facilities that may provide health services for a community. <<more needed>>

Typical actors:

* Senders – hospital administrators, hospital networks, health providers, NGOs << more>>
* Recipients – first responders, dispatch operators, emergency managers,

## Supporting Elements (non-normative)

### Common Types

<< DARRELL to REVISIT and replace with HAVE specific wording >>

Supporting Element Types borrow re-usable elements from the EDXL Common Types (ct:) that apply to and support multiple areas of the HAVE 2.0 reports, such as Location, <<MORE>> etc.. For instance incidentLocation relies on ct:EDXLLocationType, which consists of either EDXLGeoLocation for geographical information or EDXLGeoPoliticalLocation for geopolitical information. EDXLGeoLocation is of type edxl-gsf:EDXLGeoLocationType and EDXLGeoPoliticalLocation is of type ct:EDXLGeoPoliticalLocationType. This latter type consists of either a GeoCode (of type ct:ValueListType) or an Address (of type edxl-ciq:xAL:AddressType).

The following elements are used in this specification and can be found at the locations cited in the normative references in Section << REFERENCE>> above.

<< re-cast from a HAVE perspective>>

FROM TEP:

Supporting Element Types borrow re-usable elements from the EDXL Common Types (ct:) that apply to and support multiple areas of the TEP 1.0 messages, such as Location, PatientContact etc.. For instance incidentLocation relies on ct:EDXLLocationType, which consists of either EDXLGeoLocation for geographical information or EDXLGeoPoliticalLocation for geopolitical information. EDXLGeoLocation is of type edxl-gsf:EDXLGeoLocationType and EDXLGeoPoliticalLocation is of type ct:EDXLGeoPoliticalLocationType. This latter type consists of either a GeoCode (of type ct:ValueListType) or an Address (of type edxl-ciq:xAL:AddressType).

The following elements are used in this specification and can be found at the locations cited in the normative references in Section << REFERENCE>> above.

| Supporting Element/Type | Defined In |
| --- | --- |
| ct:EDXLDateTimeType | EDXL-CT (Simple Types) |
| ct:EDXLStringType | EDXL-CT (Simple Types) |
| ct:ValueListURIType | EDXL-CT (Simple Types) |
| ct:ValueType | EDXL-CT (Simple Types) |
| ct:ValueListType | EDXL-CT (Complex Types) |
| ct:ValueKeyType | EDXL-CT (Complex Types) |
| ct:EDXLGeoPoliticalLocationType | EDXL-CT (Complex Types) |
| ct:EDXLLocationType | EDXL-CT (Complex Types) |
| gsf:EDXLGeoLocationType | EDXL-GSF |
| ct:ValueListURI | EDXL-CT (Top Level Elements) |
| xal:addressType | EDXL-CIQ |
|  |  |

Some elements of the common type “ct:EDXLStringType” are denoted as [token] in the accompanying XMLper the following reference:

[token] N. Freed, XML Schema Part 2: Datatypes Second Edition, http://www.w3.org/TR/xmlschema-2/#token, W3C REC-xmlschema-2, October 2004.

The definition for token as found in the OASIS common types is: “The[value space](http://www.w3.org/TR/xmlschema-2/" \l "dt-value-space" \t "_blank)of **token** is the set of strings that do not contain the carriage return (#xD), line feed (#xA) nor tab (#x9) characters, that have no leading or trailing spaces (#x20) and that have no internal sequences of two or more spaces.”

The implication is that the XML parser will change string entries to removecarriage returns, line feeds, tab characters, leading or trailing spaces, and internal sequences of two or more spaces.

^^ END TEP:

<< TODO:

* Rationalize CT types that carried forward

Confirm that the versions of key standards (CIQ in particular) are OK.

>>

### Selecting Values from Lists

<< DARRELL to REWORD to make this HAVE-centric>>

FROM TEP:

The ValueList and ValueKey types are part of the EDXL Common Types collection. They allow standards adopters to use topic specific lists of values for elements such as <<replace with HAVE specific types>> raceEthnicity, fluenSpokenLanguages, specialTransportationNeeds, etc.. Both types have identical structure, but ValueList allows for selection of multiple values [1..\*] in the list, whereas ValueKey allows for selection of only one [1..1] value in the list.

When using a ValueList / ValueKey structure the user can specify a user-defined list by URI (either using the “urn:...” format or the more familiar “http://...” format) and then include user-defined values from that list. This structure has several advantages: (a) it provides flexibility for local communities to use community-defined terms and vocabulary; (b) it allows for the external maintenance of local or standardized lists; and (c) it avoids the problems inherent in attempting to constantly update hard-coded enumerations in a specification.

An existing vetted list should be referenced for defaults, but users could also reference their own value list .

^^ END TEP

<<TODO: clarify how the above applies (or doesn’t) to HAVE 2.0 >>

### ValueListType

<< DARRELL to REPLACE with HAVE-centric treatment of the ValueListType>>

FROM TEP:

The schema for ct:ValueListType is defined as

<xs:complexType name="ValueListType">

<xs:sequence>

<xs:element ref="ct:ValueListURI" minOccurs="1" maxOccurs="1"/>

<xs:element ref="ct:Value" minOccurs="1" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

and its application to the XML description of an element *elementName* of type ct:ValueListType would be:

<*elementName*>

<ct:ValueListURI>*valueListURI*</ct:ValueListURI>

<ct:Value>*value*\_1</ct:Value>

…

<ct:Value>*value*\_n</ct:Value>

</*elementName*>

In the Data Dictionary we describe examples of elements of type ct:ValueListType by listing value assignments to *valueListURI* and *value*\_1, …, *value*\_n.

So for instance an example for “specialMedicalNeeds” is described by

*valueListURI* = urn:myagency:gov:ahrq:specialMedicalNeeds and

*value*\_1 = Ventilator

*value*\_2 = Oxygen

which stands for

<specialMedicalNeeds>

<ct:ValueListURI>urn:myagency:gov:ahrq:specialMedicalNeeds</ct:ValueListURI>

<ct:Value>Ventilator</ct:Value>

<ct:Value>Oxygen</ct:Value>

</specialMedicalNeeds>

This example contains two special needs, one whose value is “Ventilator” and one whose value is “Oxygen”. These are notional needs created for this example. The needs are identified as values from a list whose unique Uniform Reference Identifier (URI) is “urn:myagency:gov:ahrq:specialMedicalNeeds”.

A note about ValueList: the multiplicity of ValueList can be a source for confusion. Typically, 1 is the maximum number of occurrences of ValueList. This means that at most one such list may occur for a given element; this does not preclude the user from selecting multiple entries from that list (maxOccurs = “unbounded”).

^^^ END TEP

<< TODO: replace with HAVE-centric example >>

### ValueKeyType

<< REPLACE with HAVE-centric treatment of the ValueKeyType>>

FROM TEP:

The schema for ValueKeyType is defined as

<xs:complexType name="ValueKeyType">

<xs:sequence>

<xs:element ref="ct:ValueListURI" minOccurs="1" maxOccurs="1"/>

<xs:element ref="ct:Value" minOccurs="1" maxOccurs="1"/>

</xs:sequence>

</xs:complexType>

and its application to the XML description of an element *elementName* of type ct:ValueKeyType would be:

<*elementName*>

<ct:ValueListURI>*valueListURI*</ct:ValueListURI>

<ct:Value>*value*</ct:Value>

</*elementName*>

This example uses a published list of values and definitions and selects one specific entry to describe the eyeColor of a patient:

* + *valueListURI* = urn:myagency:gov:OMG:eyeColors
  + *value* = Green

which stands for

<eyeColor>

<ct:ValueListURI>urn:myagency:gov:OMG:eyeColors</ct:ValueListURI>

<ct:Value>Green</ct:Value>

</eyeColor>

Following the approach in ValueList, we'd point ValueListURI to some other list to make a different selection of eye colors available.

^^ END TEP

<< REPLACE with HAVE-centric example >>

### EDXL Extensions

<< NEED to FILL in by explaining where extensions are used, why the mechanism is used, and when it should be used.

>>

## Element Reference Model (non-normative)

<<DIAGRAM – DARRELL to create using Sparx EA >>

## Distribution of EDXL-HAVE (non-normative)

<<NOTE: TEP (WD04) ties itself to EDXL-DE and or another messaging system due to its requirement for message metadata. Is there desire in HAVE to make this same tie? To date discussions have not indicated this.>>

HAVE messages are intended to be payloads of various messaging and/or delivery systems. Messaging systems such as EDXL-DE can treat a HAVE message as a payload. Similarly, non-message-based systems (e.g. RESTful web service) can deliver a HAVE message just as easily. An individual facility may provide an up-to-date report via a web service. An aggregator could poll the facilities that are of interest for a particular reason, or in a Publish-Subscribe scenario, subscribe to the facilities of interest.

## HAVE Elements

# Data Dictionary (normative)

FROM TEP:

The data dictionary is intended to provide detailed definition of each element contained in the EDXL-TEP standard. Where discrepancies may exist between this dictionary, the Element Reference Model (ERM), and the normative schema, the normative schema shall take precedence.

**Element** / **ElementType**– Name of the element or element type.

**Type** – Type or format of the element.

**Usage** – Optionality and Cardinality (the latter is for Element only).

If no optionality specified, then the element is “OPTIONAL”.

If no cardinality specified, the element “MUST be used once and only once”

**Definition** – Definition of the element / type.

**Comments** – Additional comments or examples to add clarity.

**Constraints** – Limits imposed on the element. Also notes the container or “parent” to which the element belongs.

**Valid Values / Examples** – A list of values that apply to this particular element, or examples which apply in order to clarify the definition. Where valid values are specified for ValueListURN/Value type pairs, these values are suggested as defaults, allowing implementations to use their own value list, or insert their own value by extending the defaults.

**Sub-elements** – List of references to elements that are part of this element

**Used In** – Source of the requirement or usage of the element.

**Requirements Supported** – A code representing and referring to each requirement contained in the original submission from the practitioner process to OASIS. EACH general, functional or information requirement is accounted for by one or more elements in the data dictionary, and/or by relationships in the message structure, one or more business rules, or through the overall standard (e.g. for general and functional requirements). Key:

gReq# - “General” requirement number.

fReq# - “Functional” requirement number.

iReq# - “Information” requirement number.

***Namespace prefixes***: we use the following prefixes for namespace scoping of elements and types

xsd = "<http://www.w3.org/2001/XMLSchema>"

predefined types in XMLSchema space

ct = "urn:oasis:names:tc:emergency:edxl:ct:1.0"

common types in EDXL space

ext = "urn:oasis:names:tc:emergency:edxl:extension:1.0"

extension mechanism for EDXL Standards

xal = "urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal"

elements / types in EDXL-CIQ-xAL (extensible Address Language) space

tep = "urn:oasis:names:tc:emergency:EDXL:TEP:1.0"

elements / types in EDXL-TEP space

tep-ct = "urn:oasis:names:tc:emergency:EDXL:TEP:Defaults:1.0"

common types in EDXL-TEP space

For an explanation of examples for ValueListType and ValueKeyType, see sections 3.2.2 and 3.2.3.

***Naming convention***: in order to mark a clear distinction between elements and types, names of elements shall not contain the string “Type” and shall be formatted in camel-type (lower case leading alpha character); types are to be terminated by the string “Type” and shall be formatted in Pascal-type (upper case leading alpha character). Acronyms that are part of a type/element identifier should preserve their all upper case format.

^^ END TEP:

<<TODO: discuss with HAVE SC about the exact format – is there a tool to generate this?>>

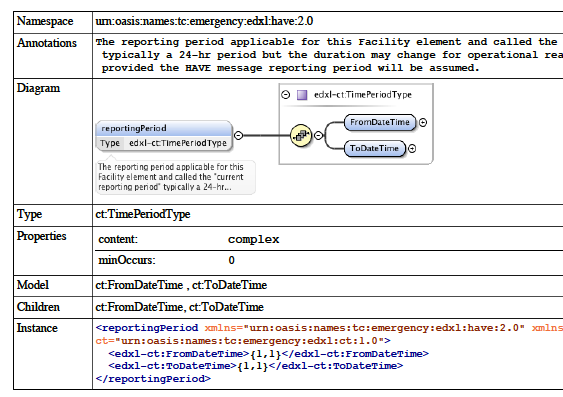
## HAVE Message

<< DARRELL to enquire of TC about the “tool” used here – hopefully we can use a known tool as opposed to doing this handraulically”>>

Example table from TEP:

|  |  |
| --- | --- |
| **Element** | **TEPMessage** |
| Type | xsd:complexType |
| Usage | REQUIRED; MUST be used once and only once |
| Definition | Group of elements used to uniquely identify a TEP message and its source. |
| Comments |  |
| Constraints |  |
| Valid Values / Examples |  |
| Sub-elements | * [messageID](#messageID|table) [1..1]: ct:[EDXLStringType](#3.2.1.Common_Types|outline) * [systemID](#systemID|table) [0..1]: ct:[EDXLStringType](#3.2.1.Common_Types|outline) * [patient](#patient|table) [1..1]: tep:[PatientType](#PatientType|table) * extension [0..\*]: ext:ExtensionType |
| Used In | EDXL-TEP |
| Requirements Supported |  |
|  |  |

The following diagram is an example of the output from OxygenXML (from XSD). It may be a starting point.



## Facility

## Services

## Resources

## Operations

## Staffing

# Conformance

The last numbered section in the specification must be the Conformance section. Conformance Statements/Clauses go here.

<< NEED the Conformance guidance for HAVE – TEP example shown below. >>

<< FROM TEP:::

An XML 1.0 element is a conforming EDXL-TEP-v1.0 Message if and only if:

a) it meets the general requirements specified in Section 4;

b) if its namespace name is "urn:oasis:names:tc:emergency:edxl:tep:1.0", and the element is valid according to the schema located at http://docs.oasis-open.org/emergency/edxl-tep-v1.0/edxl-tep-v1.0.xsd

c) if its namespace name is "urn:oasis:names:tc:emergency:edxl:tep:1.0", then its content (which includes the content of each of its descendants) meets all the additional mandatory requirements provided in the specific subsection of Section 4 corresponding to the element’s name.

**Note**: only messages that fully comply with the EDXL-TEP specification and that are complete and schematically valid, may be referred to as a “TEP Message.”.

# 

**Note: The example shown below is for informative purposes only and is a snippet of the schema – to illustrate the content.**

1. EDXL-HAVE EXAMPLE (NON-NORMATIVE)
2. Service Types and Capacities (NON-NORMATIVE)
3. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

[Participant Name, Affiliation | Individual Member]

[Participant Name, Affiliation | Individual Member]

1. Title Text

text

* 1. Subsidiary section

text

* + 1. Sub-subsidiary section

text

1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Editor** | **Changes Made** |
| [Rev number] | [Rev Date] | [Modified By] | [Summary of Changes] |