Emergency Data Exchange Language (EDXL) Tracking of Emergency Clients (TEC) Client Registry Exchange Version 1.0

Committee Specification Draft 01

13 June 2014

Specification URIs

This version:

http://docs.oasis-open.org/emergency/edxl-tec-registry/v1.0/csd01/edxl-tec-registry-v1.0-csd01.doc (Authoritative)

http://docs.oasis-open.org/emergency/edxl-tec-registry/v1.0/csd01/edxl-tec-registry-v1.0-csd01.html

http://docs.oasis-open.org/emergency/edxl-tec-registry/v1.0/csd01/edxl-tec-registry-v1.0-csd01.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/emergency/edxl-tec-registry/v1.0/edxl-tec-registry-v1.0.doc (Authoritative)

http://docs.oasis-open.org/emergency/edxl-tec-registry/v1.0/edxl-tec-registry-v1.0.html

http://docs.oasis-open.org/emergency/edxl-tec-registry/v1.0/edxl-tec-registry-v1.0.pdf

Technical Committee:

OASIS Emergency Management TC

Chair:

Elysa Jones (elysajones@yahoo.com), Individual

Editor:

Tony Mancuso (amancuso@google.com), Google Inc.

Additional artifacts:

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

- RELAX NG schema: http://docs.oasis-open.org/emergency/edxl-tec-registry/v1.0/csd01/schema/

Related work:

This specification is related to:

- *PFIF (People Finder Interchange Format) 1.4 Specification*. http://zesty.ca/pfif/1.4/.
- *Emergency Data Exchange Language (EDXL) Project Initiation Document (PID) For the PHASE II - Tracking of Emergency Clients (EDXL-TEC) Messaging Standard*. Draft Version 2.5. 07 November 2011. https://www.oasis-open.org/committees/download.php/49378/.
- *Emergency Data Exchange Language (EDXL) Requirements Statement and Draft Messaging Specification For the PHASE II - Tracking of Emergency Clients (EDXL-TEC) Messaging Standard, Version 1.5*. August 2012. https://www.oasis-open.org/committees/download.php/49379/.

Declared XML namespaces:

- http://docs.oasis-open.org/emergency/ns/edxl-tec/registry/v1.0
- urn:oasis:names:tc:emergency:edxl:tec:registry:1.0

Abstract:

This document describes a message standard for the EDXL Tracking of Emergency Clients (EDXL-TEC) Client Registry Exchange. The purpose of the standard is to provide a standard messaging format for the creation and exchange of client records in and among publicly-accessible registries to assist in tracking and repatriation of displaced individuals during emergencies, disasters, and routine day-to-day incidents. This specification is intended as a companion to an EDXL-TEC Client Tracking Exchange messaging standard, whose eventual development is anticipated.

Status:

This document was last revised or approved by the OASIS Emergency Management TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at https://www.oasis-open.org/committees/emergency/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (https://www.oasis-open.org/committees/emergency/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[edxl-tec-registry-v1.0]

Emergency Data Exchange Language (EDXL) Tracking of Emergency Clients (TEC) Client Registry Exchange Version 1.0. Edited by Tony Mancuso. 13 June 2014. OASIS Committee Specification Draft 01. http://docs.oasis-open.org/emergency/edxl-tec-registry/v1.0/csd01/edxl-tec-registry-v1.0-csd01.html. Latest version: http://docs.oasis-open.org/emergency/edxl-tec-registry/v1.0/edxl-tec-registry-v1.0.html.

Notices

Copyright © OASIS Open 2014. 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 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.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.

Table of Contents

1 Introduction

The EDXL-TEC Client Registry Exchange standard is the outgrowth of preliminary work of the EDXL Practitioner Steering Group and TEC Steering Committee, which included a *Project Initiation Document* [**EDXL-TEC PID**] and Requirements *Statement and Draft Messaging Specification* [**EDXL-TEC RQMTS**] for the Tracking of Emergency Clients Messaging Standard. Readers are encouraged to read these documents to understand the context and need for the EDXL-TEC Client Registry and Client Tracking Exchange messaging standards.

This document defines the XML message specification for the EDXL-TEC Client Registry Exchange (EDXL-TEC REGISTRY). This standard seeks to provide a client record format that emergency responders and members of the public can use to input and search client identity and location information in public registries to enable to the fullest possible finding the location of persons displaced during emergencies.

This standard seeks to encourage the creation and searching of client records by responders, family members, and other members of the public. It delegates the ranking of reliability of client record information to the persons searching client records; the client record message format does not include ranking or percentage of confidence fields. To enable persons to assess the reliability of client record data, the standard defines record metadata that identifies the source of the data entered.

The messaging standard allows the input of redundant as well as inconsistent client record information. A premise of this approach is that the more client information the better to enable those searching the registry to find people displaced during an emergency. Another premise is that much information on client identify and location during times of crisis is, of necessity, tentative or unreliable, and hence, a registry is most effective when it enables the collection of as much client information as possible to aid those searching the registry to "triangulate" the location of a displaced person from all available client record information (based on the searcher's assessment of the reliability of the source of the retrieved client record information).

Another criteria considered during the definition of this standard is the need for interoperability among registries. To facilitate interoperability, client record information is defined in as general a format as possible, allowing basic client information together with free-form text notes to be exchanged among registries. This approach obviates the need for each registry to conform to idiosyncratic or complicated client record field format requirements. It also facilitates the creation of simple client registry user interfaces to aid in the quick design and deployment of web-based client registry UIs in response to a disaster.

This EDXL standard incorporates [**PFIF**], People Finder Interchange Format, version 1.4. PFIF is an open standard that has been implemented and used during emergencies to track displaced persons. Goals of incorporating PFIF, with any additional enhancements, into this EDXL standard include: 1) to facilitate the governance and implementation of a standard for the broad-based exchange of data between person-based registry systems, and 2) to improve processes for search, people-finding, and family re-unification across existing registry systems supported by private industry, NGO's, and federal, state, local, and tribal entities.

1.1 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].

1.2 Normative References

[**EDXL-TEC RQMTS**]   Emergency Data Exchange Language (EDXL) Requirements Statement and Draft Messaging Specification, August 2012 https://www.oasis-open.org/committees/download.php/49379/EDXL-TEC-RqmtsdraftMessagingSpec-DRAFT%20v15.doc

[EDXL-TEC PID]   Emergency Data Exchange Language (EDXL) Project Initiation Document (PID), 11/07/2011 https://www.oasis-open.org/committees/download.php/49378/EDXL-TEC%20Project%20Initiation%20Document%20(PID)v2.5.pdfhttps://www.oasis-open.org/apps/org/workgroup/emergency-tec/download.php/49378/EDXL-TEC Project Initiation Document (PID)v2.5.pdf

[PFIF] People Finder Interchange Format) 1.4 Specification, May 29, 2012 http://zesty.ca/pfif/1.4/

[**RFC822**]   Crocker, David H., Standard for the Format of ARPA Internet Text Messages," August 13, 1982 http://www.ietf.org/rfc/rfc0822.txt

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.

[**RFC3339**]   Klyne, G, et. al, "Date and Time on the Internet: Timestamps," RFC 3339, July 2002 http://www.ietf.org/rfc/rfc3339.txt

2 Design Goals and Principles

- The goal of the EDXL-TEC Client Registry Exchange messaging standard is to bring people and data together during a crisis. In effect, the standard aims to promote convergence: that is:
  - the convergence of people who seek the same person
  - convergence of information about a person obtained from various sources
  - convergence of duplicated data; and,
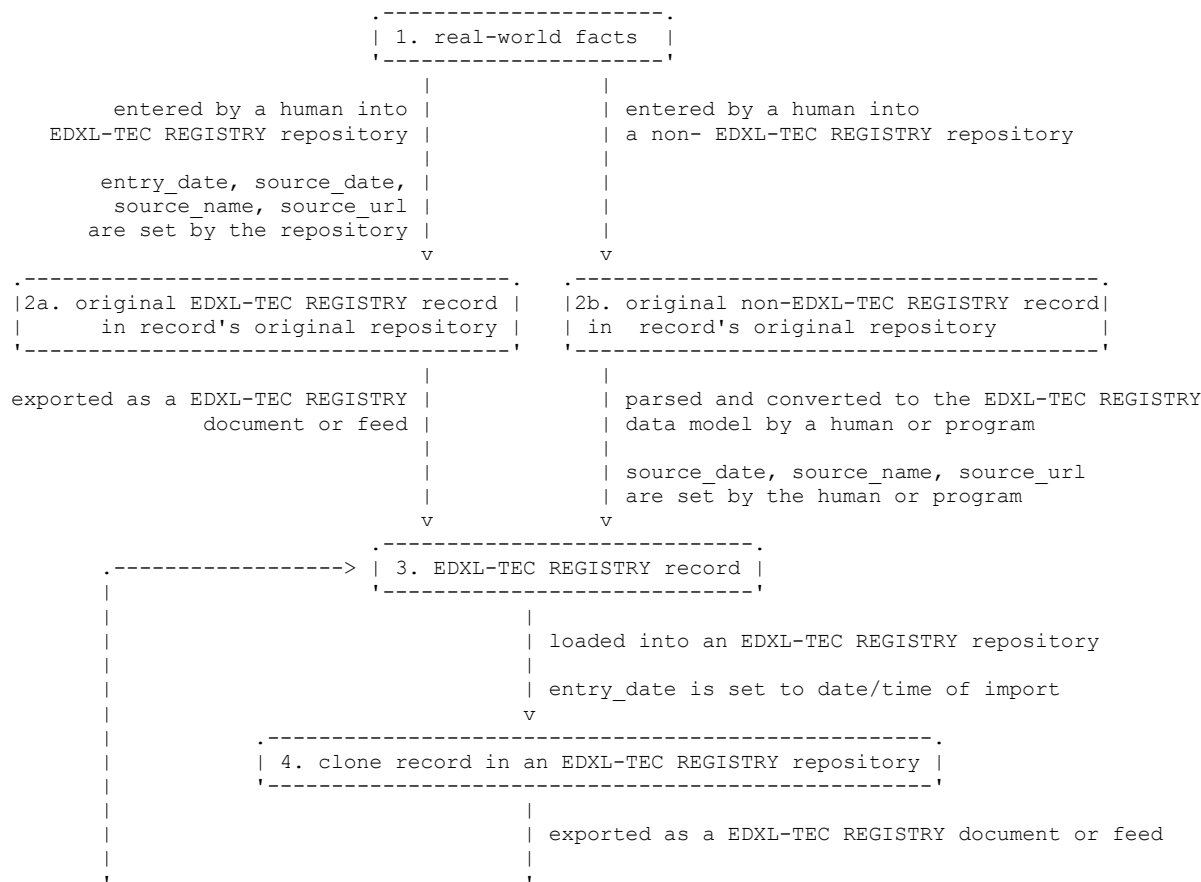 - ultimately, convergence of missing people with their loved ones.

To aid such convergence:

- Client record data should be traceable. Since data comes from sources of unknown reliability and accountability, information on the origins of data should be maintained to help users ascertain its trustworthiness.
- Each user (including data aggregators) has a unique perspective on the world and should be responsible for choosing which data sources to trust. It is not possible to dictate truths about all data from a central authority.
- Because multiple records may refer to the same person, records can be associated with each other. But, since there is no central authority on the trustworthiness of client record data, each aggregator should make its own decisions about which records to associate.
- Each record belongs to an *original repository*, which is the repository where the record was first entered (which may or may not be a registry compliant with this messaging standard). A client record may be copied to other places, but only the original repository should be allowed to change the contents of a record.
- It should be possible to resolve multiple copies of the same record that have been imported via different data paths.
- All dates and times should be recorded in client records as UTC, not in local time, since records may be exchanged between repositories in different time zones. This standard adopts the use of dates and times in [**RFC3339**] format (with only UTC allowed). User interfaces can convert UTC time to local time for display to the user.

3 Data Life Cycle

Each EDXL-TEC Client Registry Exchange (EDXL-TEC REGISTRY) repository MAY contain *original records* and *clone records*. An original record is a record residing in its original repository (the registry in which it was created); a clone record is a copy of a record that originated in another repository. The following diagram describes the life of a EDXL-TEC REGISTRY record as it is created and then travels to other repositories.

Each EDXL-TEC Client Registry Exchange (EDXL-TEC REGISTRY) repository MAY contain *original records* and *clone records*. An original record is a record residing in its original repository (the registry in which it was created); a clone record is a copy of a record that originated in another repository. The following diagram describes the life of a EDXL-TEC REGISTRY record as it is created and then travels to other repositories.

```
                              .---------------------.
                              | 1. real-world facts |
                              '---------------------'
                                   |              |
             entered by a human into |            | entered by a human into
        EDXL-TEC REGISTRY repository |            | a non- EDXL-TEC REGISTRY repository
                                   |              |
             entry_date, source_date, |          |
               source_name, source_url |         |
             are set by the repository |         |
                                  v              v
.-------------------------------------.  .--------------------------------------.
|2a. original EDXL-TEC REGISTRY record |  |2b. original non-EDXL-TEC REGISTRY record|
|     in record's original repository |  | in  record's original repository       |
'-------------------------------------'  '--------------------------------------'
                                   |              |
exported as a EDXL-TEC REGISTRY |                | parsed and converted to the EDXL-TEC REGISTRY
            document or feed |                    | data model by a human or program
                                   |              |
                                   |              | source_date, source_name, source_url
                                   |              | are set by the human or program
                                   v              v
                              .------------------------------.
        .-----------------> | 3. EDXL-TEC REGISTRY record |
        |                     '------------------------------'
        |                                 |
        |                                 | loaded into an EDXL-TEC REGISTRY repository
        |                                 |
        |                                 | entry_date is set to date/time of import
        |                                 v
        |             .-----------------------------------------------------.
        |             | 4. clone record in an EDXL-TEC REGISTRY repository |
        |             '-----------------------------------------------------'
        |                                 |
        |                                 | exported as a EDXL-TEC REGISTRY document or feed
        |                                 |
        '---------------------------------'
```

3.1 Incremental Export Mechanism

Whenever an EDXL-TEC REGISTRY repository adds a new original record or clone record, it MUST set the entry_date field to the current time. This time value MUST never decrease as records are added. A client can incrementally update its copy of a repository by querying for all records with an entry_date greater than or equal to the entry_date of the last received record.

3.2 Data Update Mechanism

The original repository for a record (2a or 2b in the diagram above) can update any of the fields on a record after it is created, except the `person_record_id` field. Whenever an EDXL-TEC REGISTRY repository creates or updates an original record, it MUST set both the `source_date` and `entry_date` fields to the current time. When a repository imports a EDXL-TEC REGISTRY record that has the same record identifier as an existing record, it should keep the version with the latest `source_date`.

3.3 Data Expiry Mechanism

If present, the `expiry_date` field indicates when a record should be deleted to preserve the privacy of the personal information it contains. Conforming EDXL-TEC REGISTRY implementations MUST meet the following requirements:

- Within one day after `expiry_date`, an EDXL-TREC CRE repository MUST make the contents of the person record and any associated note records inaccessible to all external clients, including users and machine API clients.
- Thereafter, if the repository exports its data through an API, it should continue to export a placeholder record in the place of the expired person record. This placeholder should keep the same `person_record_id` and `expiry_date` values, and have both `source_date` and `entry_date` set to the time that the placeholder was created. All other fields should be empty or omitted.
- Within 60 days after `expiry_date`, an EDXL-TEC REGISTRY repository MUST permanently and unrecoverably delete all its copies (including backups) of the contents of the person record and any associated note records except for the `person_record_id`, `source_date`, `entry_date`, and `expiry_date` fields needed to produce the placeholder.

To satisfy a user request to delete an existing original record, an EDXL-TEC REGISTRY repository should set the record's `expiry_date` to the current time. (In accordance with the preceding section, it would also set the `source_date` and `entry_date` to the current time.) The expiry mechanism described above would then cause the deletion to propagate to other conforming EDXL-TEC REGISTRY repositories.

4 Data Model

There are two types of EDXL-TEC REGISTRY records.

1. Person records – information that identifies a person. Person records may be created by those who seek a displaced person and by those who have information on a displaced person. The person record for a person is the point of convergence for all parties;
2. Note records – information about the current status of a displaced person. Each note record belongs to a particular person, and a person record may have with any number of associated note records. The note records on a person are the growing pool of shared knowledge about a person.

An EDXL-TEC REGISTRY person record should be updated only if the information in the record is incorrect. If the status or location of a particular person has changed, this should be indicated by adding a new note record associated with that person record.

4.1 Person Records

A person record contains 25 fields. There may be multiple person records for the same person. In fact, any given application that imports data from multiple sources is likely to acquire multiple person records for the same person. It is up to the application to associate such records (see Suggested Relational Database Schema, below). It is recommended that applications keep copies of all records, and separately keep track of records that correspond to the same person.

4.1.1 Person Record Metadata

This metadata is necessary to enable users of the data to trace and ascertain the reliability of the person record data.

person_record_id (ASCII string, required)

> Unique identifier for this record, which consists of a lowercase ASCII domain name followed by a slash and a local identifier. The domain name identifies this record's original repository, which is the authority for this record. The format of the local identifier is up to the original repository. When the person_record_id begins with a domain other than the application's own domain, it means this record is a clone record.

entry_date (ASCII string in the form "yyyy-mm-ddThh:mm:ssZ", optional):

> Date in UTC that this copy of this record was stored (see Incremental Export Mechanism, above).

expiry_date (ASCII string in the form "yyyy-mm-ddThh:mm:ssZ", optional):

> Date in UTC after which this record should be deleted (see Data Expiry Mechanism, above).

author_name (Unicode string, optional):

> The full name of the person who entered this record.

author_email (e-mail address string, optional):

> The preferred contact e-mail address of the person who entered this record.

author_phone (ASCII string, optional):

> The preferred contact phone number of the person who entered this record.

source_name (Unicode string, optional):

> The human-readable name of the original repository of this record.

source_date (ASCII string in the form "yyyy-mm-ddThh:mm:ssZ", required):

> The date in UTC that the original copy of this record was created in its original repository.

source_url (URL string, optional):

The URL to this record in its original repository (as specific as possible, down to the URL of the individual record).

4.1.1.1 Person Record Data

These fields contain information that is used to identify a person; this is information that is not expected to change unless it is incorrect. Searches for person records should search over these fields.

full_name (Unicode string, required):

> The full name of the person sought or found, combined in the order and fashion customary to the person, language, and culture. For example, a typical English name would be formatted as a first, middle, and last name with spaces between them, whereas a typical Chinese name would be formatted with the family name first and no spaces between characters. If the person has more than one full name (for example, both an English name and a Chinese name), begin with the most commonly used full name and use newline (Unicode U+000A) characters to separate the full names.

given_name (Unicode string, optional):

> Given name of the person sought or found, optionally followed by a space and any middle names or middle initials.

family_name (Unicode string, optional):

> Family name of the person sought or found.

alternate_names (Unicode string, optional):

> Any other names associated with the person that are not part of the person's usual full name, such as nicknames, alternate spellings, and transliterations. For example, hiragana readings of Japanese kanji could go in this field. Use newline (Unicode U+000A) characters to separate names.

description (Unicode string, optional):

> Description of the person in free-form text. In entry forms, a multi-line text box is appropriate for this field.

sex (ASCII string, optional):

> Physical sex of the person sought or found, specified as one of the three strings `female`, `male`, or `other`. If the sex is unknown, omit this field.

date_of_birth (ASCII string in the form "yyyy", "yyyy-mm", or "yyyy-mm-dd", optional):

> Exact or approximate date of birth of the person sought or found.

age (integer, or ASCII string in the form "min-max", optional):

> Approximate age of the person sought or found, in years since birth as of the source_date of this record. The value of this field is either a single decimal integer or an inclusive range given as two decimal integers separated by a hyphen. This field has no defined meaning when source_date is missing.

home_street (Unicode string, optional):

> Street name of the home address of the person sought or found. To protect user privacy, applications should generally avoid including a street number in this field.

home_neighborhood (Unicode string, optional):

> Name of the home neighborhood of the person sought or found. Use this field for the names of official or unofficial geographic regions not captured by the other address fields.

home_city (Unicode string, optional):

> Home city of the person sought or found.

home_state (Unicode string, optional):

> Home state, province, territory, district, region, parish, county, or department of the person sought or found, specified as an uppercase ISO 3166-2 code (preferred) or by its name.

home_postal_code (ASCII string, optional):

> Postal code of the home address of the person sought or found, in the format most commonly used in the country.

home_country (ASCII ISO-3166-1 country code, optional):

> Home country of the person sought or found, specified as an uppercase two-letter ISO-3166-1 country code.

photo_url (URL string, optional):

> URL to an image of an identifying photograph of the person sought or found.

profile_urls (URL string, optional):

> URLs to the person's profile pages on other websites. Use newline (Unicode U+000A) characters to separate URLs.

4.2 Note Records

Each note record belongs to exactly one person record. There may be any number of note records associated with a particular person record. (See below for implementation notes. For example, a database may implement this by including a foreign key, person_record_id, that refers to the person record. An object-oriented representation might implement this by embedding a list of note objects within the person object.)

Note records are used to provide updated, current information on a missing person. Every note has a timestamp and information on the author of the note. Applications can use the timestamp to determine the most recent value of a given field. Users can use the author information to ascertain the reliability of a given field.

4.2.1.1 Note Record Metadata

note_record_id (ASCII string, required):

> Unique identifier for this record, which consists of a domain name followed by a slash and a local identifier. The domain name identifies this record's home repository, which is the authority for this record. The format of the local identifier is up to the home repository. When the note_record_id begins with a domain other than the application's own domain, it means this record is a clone of a record from another source.

person_record_id (ASCII string, required):

> The person_record_id of the person record to which this note belongs.

linked_person_record_id (ASCII string, optional):

> The person_record_id of another person record to associate with the record to which this note belongs. When this field is present, it signifies that the author of this note believes that the two records identified by person_record_id and linked_person_record_id refer to the same person. If this field is present, the text field should explain how these records were determined to refer to the same person.

entry_date (ASCII string in the form "yyyy-mm-ddThh:mm:ssZ", optional):

> Date in UTC that this copy of this record was stored. An EDXL-TEC REGISTRY repository must guarantee that this value increases monotonically as records are added, so that a client can update a copy of a repository by querying for all records with an entry_date greater than or equal to the entry_date of the last received record.

author_name (Unicode string, required):

> The full name of the person who entered this note.

author_email (e-mail address string, optional):

> The preferred contact e-mail address of the person who entered this note.

author_phone (ASCII string, optional):

The preferred contact phone number of the person who entered this note.

source_date (ASCII string in the form "yyyy-mm-ddThh:mm:ssZ", required):

The date in UTC that the original copy of this note was created in its home repository. In most cases, notes should be sorted by this field for display.

4.2.1.2 Note Record Data

The `author_made_contact`, `status`, `email_of_found_person`, `phone_of_found_person` and `last_known_location` fields store data that changes over time. When these fields are present in a note record, the record is specifying new values for these fields, and the source_date field indicates the date that the new values took effect. So, for example, an application that wants to display the most recent known location can look for the note with the latest source_date that has a non-empty `last_known_location` field.

author_made_contact (ASCII string, optional):

This value is the string `true` if the author of this note has personally contacted the missing person, or `false` otherwise. If this field is `true`, the text field of this note should describe HOW and WHEN the person was contacted or seen.

status (ASCII string, optional)

Status of the person sought or found, specified as one of the following five strings:

information_sought

The author of the note is seeking information on the person in question.

is_note_author

The author of the note is the person in question.

believed_alive

The author of the note has received information that the person in question is alive.

believed_missing

The author of the note has reason to believe that the person in question is still missing.

believed_dead

The author of the note has received information that the person in question is dead.

email_of_found_person (e-mail address string, optional):

The current preferred contact e-mail address of the FOUND person. This field is present ONLY if the person has been FOUND. If this field is present, the text field of this note should describe HOW the person's contact information was determined.

phone_of_found_person (ASCII string, optional):

The current preferred contact phone number of the FOUND person. This field is present ONLY if the person has been FOUND. If this field is present, the text field of this note should describe HOW the person's contact information was determined.

last_known_location (Unicode string, optional):

A free-form description of the last known location of the person, which can be expressed as geographical coordinates. To specify geographic coordinates in this field, give the latitude in decimal degrees (positive for north), then a comma, then the longitude in decimal degrees (positive for east). If this field is present, the text field of this note (the next field listed below) should describe HOW the person's location was determined.

text (Unicode string, required):

Free-form text description of the person's current condition, situation and location details, where they were last seen, corrections to other information, and so on. In entry forms, a multi-line text box is appropriate for this field.

photo_url (URL string, optional):

URL to an image to include with this note.

5 XML Format Specification

The XML Namespace for EDXL-TEC REGISTRY is:

- http://docs.oasis-open.org/emergency/ns/edxl-tec/registry/v1.0http://docs.oasis-open.org/emergency/ns/edxl-tec/registry/v1.0

The MIME type for an EDXL-TEC Client Registry Exchange document is:

- `application/edxl-tec-registry+xml`

A valid EDXL-TEC REGISTRY XML document consists of a single `edxl-tec-registry` element containing one or more person or note elements, each of which contains child elements for the fields described above. In a person element, the `person_record_id`, `source_date`, and `full_name` fields are mandatory. In a note element, the `note_record_id`, `author_name`, and `source_date` fields are mandatory. All other fields are optional. The order of the child elements within a person or note element is not significant.

A note element can exist inside or outside a person element. When a note element appears outside a person element, the note must contain a `person_record_id`. Otherwise, the `person_record_id` field is optional, and if present, must match the `person_record_id` of the enclosing person.

The RELAX NG Schema for EDXL-TEC REGISTRY, given in RELAX NG Compact Syntax, is as follows:

```
namespace edxl-tec-registry = "http://docs.oasis-open.org/emergency/ns/edxl-
tec/registry/v1.0"

start = element edxl-tec-registry:edxl-tec-registry { person* & note* }

person = element edxl-tec-registry:person {
    element edxl-tec-registry:person_record_id { record_id } &
    element edxl-tec-registry:entry_date { time } ? &
    element edxl-tec-registry:expiry_date { time } ? &
    element edxl-tec-registry:author_name { text } ? &
    element edxl-tec-registry:author_email { email } ? &
    element edxl-tec-registry:author_phone { phone } ? &
    element edxl-tec-registry:source_name { text } ? &
    element edxl-tec-registry:source_date { time } &
    element edxl-tec-registry:source_url { url } ? &
    element edxl-tec-registry:full_name { text } &
    element edxl-tec-registry:given_name { text } ? &
    element edxl-tec-registry:family_name { text } ? &
    element edxl-tec-registry:alternate_names { text } ? &
    element edxl-tec-registry:description { text } ? &
    element edxl-tec-registry:sex { sex } ? &
    element edxl-tec-registry:date_of_birth { approx_date } ? &
    element edxl-tec-registry:age { approx_age } ? &
    element edxl-tec-registry:home_street { text } ? &
    element edxl-tec-registry:home_neighborhood { text } ? &
    element edxl-tec-registry:home_city { text } ? &
    element edxl-tec-registry:home_state { text } ? &
    element edxl-tec-registry:home_postal_code { text } ? &
    element edxl-tec-registry:home_country { country_code } ? &
    element edxl-tec-registry:photo_url { url } ? &
    element edxl-tec-registry:profile_urls { text } ? &
    note*
}

note = element edxl-tec-registry:note {
```

```
        element edxl-tec-registry:note_record_id { record_id } &
        element edxl-tec-registry:person_record_id { record_id } ? &
        element edxl-tec-registry:linked_person_record_id { record_id } ? &
        element edxl-tec-registry:entry_date { time } ? &
        element edxl-tec-registry:author_name { text } &
        element edxl-tec-registry:author_email { email } ? &
        element edxl-tec-registry:author_phone { phone } ? &
        element edxl-tec-registry:source_date { time } &
        element edxl-tec-registry:author_made_contact { boolean } ? &
        element edxl-tec-registry:status { status } ? &
        element edxl-tec-registry:email_of_found_person { email } ? &
        element edxl-tec-registry:phone_of_found_person { phone } ? &
        element edxl-tec-registry:last_known_location { text } ? &
        element edxl-tec-registry:text { text } &
        element edxl-tec-registry:photo_url { url } ?
}

record_id = xsd:string { pattern = ".+/.+" }
time = xsd:dateTime { pattern = "\d\d\d\d-\d\d-\d\dT\d\d:\d\d:\d\d(\.\d+)?Z"
}
email = xsd:string { pattern = ".+@.+" }
phone = xsd:string { pattern = "[\-+()\d ]+" }
url = text
sex = "female" | "male" | "other"
approx_date = xsd:string { pattern = "\d\d\d\d(-\d\d(-\d\d)?)?" }
approx_age = xsd:string { pattern = "\d+(-\d+)?" }
country_code = xsd:string { pattern = "[A-Z][A-Z]" }
boolean = "true" | "false"
status = "information_sought" | "is_note_author" |
    "believed_alive" | "believed_missing" | "believed_dead"
```

6 Conformance

An implementation claiming to conform to the requirements of this specification must be able to produce a valid EDXL-TEC REGISTRY XML message consisting of at least all the mandatory fields as specified in Section 5.

Appendix A. Atom Feed Specifications

EDXL-TEC Client Registry Exchange (EDXL-TEC REGISTRY) XML documents can be embedded into Atom 1.0 feeds. The EDXL-TEC REGISTRY document should be embedded using an XML namespace and inserted as an immediate child of the entry element.

Atom 1.0 defines a top-level feed element that contains any number of entry elements. The top-level element should declare the EDXL-TEC REGISTRY namespace. The recommended prefix is `edxl-tec-registry`, so the top-level element should look like this:

```
<feed xmlns="http://www.w3.org/2005/Atom"

      xmlns:edxl-tec-registry="http://docs.oasis-open.org/emergency/ns/edxl-tec/registry/v1.0">

...

</feed>
```

The rest of this section offers recommendations on how applications should populate the standard Atom elements so that the feed will make sense to existing feed-reading software. Nonetheless, the embedded EDXL-TEC REGISTRY document takes precedence over any redundant information that appears in Atom elements.

Two kinds of EDXL-TEC REGISTRY Atom feeds are defined here: *person feeds* in which each entry contains a person, and *note feeds* in which each entry contains a note. A person feed is roughly analogous to a blog feed containing blog entries; a note feed is roughly analogous to a comment feed on a particular blog entry. For example, one application might subscribe to a person feed in order to aggregate missing person records from other databases; another application might subscribe to a note feed in order to display a stream of notes with updates about a particular person.

A.1 Atom Person Feeds

An Atom person feed provides at least the following elements within the feed element:

id

> This element should contain a unique URI associated with this feed. This might be the URL to the website that corresponds to the database or service providing this feed.

title

> This element should contain the name of this feed. This should include the title of the database or service providing this feed.

subtitle

> This element should contain a phrase or sentence describing this feed. This would be the place to explain how this feed is produced, for example: "Scraped daily by FooMatic 2.3 from http://example.org/".

updated

> This element should contain the date and time in UTC that this feed was last updated, given in "yyyy-mm-ddThh:mm:ssZ" format.

link

> This element should contain a URL from which this feed can be retrieved. This element should have a `rel` attribute whose value is `self`.

An Atom person feed provides at least the following elements within each entry element:

edxl-tec-registry:person

> This element contains child elements for the fields of the person record, as well as zero or more edxl-tec-registry:note elements. A service wishing to provide a complete export would include all the note records associated with the person here.

id

> This element should contain a URI string consisting of the scheme " edxl-tec-registry:" followed by the value of the person_record_id field.

title

> This element should contain the value of the full_name field.

author

> This element should contain a name element containing the value of the author_name field and an email element containing the value of the author_email field in the person record.

updated

> This element should contain the value of the source_date field in the person record.

content

> This element should contain a human-readable HTML formatting of the information in the person record. It is up to the application to decide how to format the content.

source

> This element should contain a copy of the title element of this feed. This element may also contain copies of any other child elements of the feed element.

A.2 Atom Note Feeds

An Atom note feed provides at least the following elements within the feed element:

id

> This element should contain a unique URI associated with this feed. This might be the URL to the website that corresponds to the database or service providing this feed.

title

> This element should contain the name of this feed. This should include the title of the database or service providing this feed, followed by a more specific title that describes how the notes were selected from the database or service. For example, for a note feed about a particular person, the title could be the title of the service followed by the first name and last name of the person in question.

subtitle

> This element should contain a phrase or sentence describing this feed. This would be the place to explain how this feed is produced, for example: "Exported by CiviCRM 1.1, http://www.example.org/."

updated

> This element should contain the date and time in UTC that this feed was last updated, given in "yyyy-mm-ddThh:mm:ssZ" format.

link

> This element should contain a URL from which this feed can be retrieved. This element should have a `rel` attribute whose value is `self`.

An Atom note feed provides at least the following elements within each entry element:

edxl-tec-registry:note

> This element contains child elements for the fields of the note record.

id

> This element should contain a URI string consisting of the scheme "edxl-tec-registry:" followed by the value of the note_record_id field.

title

> This element should contain an excerpt of the text field.

author

This element should contain a name element containing the value of the author_name field and an email element containing the value of the author_email field in the note record.

updated

This element should contain the value of the source_date field in the note record.

content

This element should contain an HTML formatting of the text field in the note record. It is up to the application to decide how to format the content.

Appendix B. RSS Feed Specifications

EDXL-TEC-Client Registry Exchange (EDXL-TEC-REGISTRY) documents can be embedded into RSS 2.0 feeds. (In RSS 2.0 terminology, this section defines an RSS 2.0 module.) The EDXL-TEC-REGISTRY document should be specified using an XML namespace and embedded as an immediate child of the item element.

RSS 2.0 defines two main elements, channel and item, that are enclosed in a top-level RSS element. The top-level element should declare the EDXL-TEC-REGISTRY namespace. The recommended prefix is `edxl-tec-registry`, so the top-level element should look like this:

```
<rss version="2.0" xmlns: edxl-tec-registry="http://docs.oasis-
open.org/emergency/ns/edxl-tec/registry/v1.0">

...

</rss>
```

The rest of this section offers recommendations on how applications should populate the standard RSS elements so that the feed will make sense to existing feed-reading software. Nonetheless, the embedded EDXL-TEC-REGISTRY document takes precedence over any redundant information that appears in RSS elements.

Two kinds of EDXL-TEC-REGISTRY RSS feeds are defined here: person feeds in which each item contains a person, and note feeds in which each item contains a note.

B.1 RSS Person Feeds

An RSS person feed provides at least the following elements within the channel element:

title

> This element should contain the name of this feed, which should include the title of the database or service providing this feed.

description

> This element should contain a phrase or sentence describing this feed. This is the place to explain how this feed is produced, for example: "Scraped daily by FooMatic 2.3 from http://example.org/".

lastBuildDate

> This element should contain the date and time in UTC that this feed was last updated, given in [**RFC822**] date format, for example: "Fri, 16 Aug 2013 00:00:01 GMT".

link

> This element should contain a URL to the website that corresponds to the database or service providing this feed.

An RSS person feed provides at least the following elements within each item element:

edxl-tec-registry:person

> This element contains child elements for the fields of the person record, as well as zero or more edxl-tec-registry:note elements. A service wishing to provide a complete export would include all the note records associated with the person here.

guid

> This element should contain the value of the person_record_id field.

title

> This element should contain the value of the full_name field.

author

> This element should contain the value of the author_email field, followed by a space and the value of the author_name field enclosed in parentheses.

pubDate

> This element should contain the date in the source_date field in the person record, converted to [**RFC822**] date format, for example: "Fri, 16 Aug 2013 00:00:01 GMT". The timezone MUST be GMT and the year MUST have four digits.

description

> This element should contain a human-readable HTML formatting of the information in the person record. It is up to the application to decide how to format the description.

source

> This element should contain the value of the source_name field.

link

> This element should contain the value of the source_url field.

B.2 RSS Note Feeds

An RSS note feed provides at least the following elements within the channel element:

title

> This element should contain the name of this feed. This should include the title of the database or service providing this feed, followed by a more specific title that describes how the notes were selected from the database or service. For example, for a note feed about a particular person, the title could be the title of the service followed by the first name and last name of the person in question.

description

> This element should contain a phrase or sentence describing the feed. This is the place to explain how the feed is produced, for example: "Scraped daily by FooMatic 2.3 from http://www.example.org/".

lastBuildDate

> This element should contain the date and time in UTC that this feed was last updated, given in [**RFC822**] date format, for example: "Fri, 16 Aug 2013 00:00:01 GMT".

link

> This element should contain a URL to the website that corresponds to the database or service providing this feed. For a note feed about a particular person, this link could point to the web page for that person's record.

An RSS note feed provides at least the following elements within each item element:

edxl-tec-registry:note

> This element contains child elements for the fields of the note record.

guid

> This element should contain the value of the note_record_id field.

author

> This element should contain the value of the author_email field, followed by a space and the value of author_name field enclosed in parentheses.

pubDate

> This element should contain the date in the source_date field in the note record, converted to [**RFC822**] date format, for example: "Fri, 16 Aug 2013 00:00:01 GMT". The timezone MUST be GMT and the year MUST have four digits.

description

This element should contain an HTML formatting of the text field in the note record. It is up to the application to decide how to format the description.

Appendix C. Suggested Relational Database Schema

This section suggests a possible relational database schema for storing EDXL-TEC-Client Registry Exchange (EDXL-TEC-REGISTRY) data. The exact details of a database design are up to each application; this is one possible starting point. A relational database could store EDXL-TEC-REGISTRY records in two tables, person and note, for the two types of records.

```
PERSON table:
    string      person_record_id            primary key
    datetime    entry_date
    datetime    expiry_date
    string      author_name
    string      author_email
    string      author_phone
    string      source_name
    datetime    source_date
    string      source_url
    string      full_name
    string      given_name
    string      family_name
    string      alternate_names
    text        description
    string      sex
    string      date_of_birth
    string      age
    string      home_street
    string      home_neighborhood
    string      home_city
    string      home_state
    string      home_postal_code
    string      home_country
    string      photo_url
    string      profile_urls

NOTE table:
    string      note_record_id              primary key
    string      person_record_id            foreign key not null
    string      linked_person_record_id     foreign key or null
    datetime    entry_date
    string      author_name
    string      author_email
    string      author_phone
    datetime    source_date
    boolean     author_made_contact
    string      status
    string      email_of_found_person
    string      phone_of_found_person
    string      last_known_location
    text        text
    string      photo_url
```

To link a foreign person record with a local person record, the application adds a note associated with the local person record, with a `linked_person_record_id` field containing the `person_record_id` of the foreign record. The other fields of the note describe the circumstances of the decision to merge: `source_date` indicates the date of the decision, text gives the reason for the decision, and `author_name` names the person, program, or other entity that made the decision. This specification

does not dictate how an application would decide whether to merge two records; a merge could be initiated by a human operator or by a software algorithm that look for records with similar data. Recording the merge decision in a note record makes it possible to back out of a bad merge decision, and recording the name of the person or program in the `author_name` field makes it possible to track down the cause of an incorrect merge.

When displaying a person record, the application can then look for all the non-empty `linked_person_record_id` fields among the notes that belong to that person record, and display all the linked records or a merged view of the linked records.

Appendix D. Acknowledgments

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

Participants:

- Ka-Ping Yee, Google Individual Member, author of PFIF, Person Finder Interchange Format, which forms the basis of this EDXL-TEC Client Registry Exchange Specification
- Darrell O'Donnell, Individual Member.
- Anthony Mancuso, Google Individual Member, editor.

Appendix E. Revision History

Revision	Date	Editor	Changes Made
wd01	8/15/2013	Tony Mancuso	Created initial draft by from PFIF 1.4 Specification.
Wd02	11/20/13	Tony Mancuso	Added comments from EM-TEC SC
wd03	12/11/13	Tony Mancuso	Added PFIF background in introduction; minor edits/cleanup
wd04	2/11/134	Tony Mancuso	Added comments and changes resulting from review by OASIS EM-TC.
wd05	3/1/14	Tony Mancuso	Added language to description of last_known_location field to clarify that the field can include geolocation coordinates and the following text field is used to add information on how last_known_location determined.
wd06	3/15/14	Tony Mancuso	Minor formatting changes. Version approved on 4/8/14 by EM-TC for submission to admin for public review.
wd07	6/9/14	Tony Mancuso	Added Conformance information in Section 6.
wd08	7/3/14	Tony Mancuso	Fixed links; style changes