# Emergency Data Exchange Language (EDXL) Common Types (CT) Version 1.0

## Committee Specification Draft 02

## 22 November 2011

### Specification URIs

**This version:**
http://docs.oasis-open.org/emergency/edxl-ct/v1.0/csd02/edxl-ct-v1.0-csd02.odt (Authoritative)
http://docs.oasis-open.org/emergency/edxl-ct/v1.0/csd02/edxl-ct-v1.0-csd02.html
http://docs.oasis-open.org/emergency/edxl-ct/v1.0/csd02/edxl-ct-v1.0-csd02.pdf

**Previous version:**
http://docs.oasis-open.org/emergency/edxl-ct/v1.0/csd01/edxl-ct-v1.0-csd01.odt (Authoritative)
http://docs.oasis-open.org/emergency/edxl-ct/v1.0/csd01/edxl-ct-v1.0-csd01.html
http://docs.oasis-open.org/emergency/edxl-ct/v1.0/csd01/edxl-ct-v1.0-csd01.pdf

**Latest version:**
http://docs.oasis-open.org/emergency/edxl-ct/v1.0/edxl-ct-v1.0.odt (Authoritative)
http://docs.oasis-open.org/emergency/edxl-ct/v1.0/edxl-ct-v1.0.html
http://docs.oasis-open.org/emergency/edxl-ct/v1.0/edxl-ct-v1.0.pdf

**Technical Committee:**
OASIS Emergency Management TC

**Chair:**
Elysa Jones (elysajones@yahoo.com), Individual

**Editors:**
Werner Joerg (Werner.Joerg@iem.com), IEM, Inc.
Rex Brooks (rex.brooks@ncoic.org), Network Centric Operations Industry Consortium
Jeff Waters (jeff.waters@navy.mil), US Department of Defense (DoD)
Don McGarry (dmcgarry@mitre.org), MITRE Corp

**Additional artifacts:**
This prose specification is one component of a Work Product which also includes:
- XML schema: http://docs.oasis-open.org/emergency/edxl-ct/v1.0/csd02/xsd/

**Declared XML namespaces:**
- http://docs.oasis-open.org/ns/emergency/edxl-ct/v1.0
- urn:oasis:names:tc:emergency:edxl:ct:1.0
- urn:oasis:names:tc:emergency:EDXL:CT:1.0

**Abstract:**
EDXL Common Types describes components and component types that can be reused across the suite of Emergency Data Exchange Language (EDXL) standards. These common components and types are intended for internal use by the Emergency Management Technical Committee and its subcommittees as they develop specific standards utilizing these types.

**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 Work Product 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 http://www.oasis-open.org/committees/emergency/.

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

**Citation format:**

When referencing this Work Product the following citation format should be used:

**[EDXL-CT]**

*Emergency Data Exchange Language (EDXL) Common Types (CT) Version 1.0*. 22 November 2011. OASIS Committee Specification Draft 02. http://docs.oasis-open.org/emergency/edxl-ct/v1.0/csd02/edxl-ct-v1.0-csd02.html.

# Notices

Copyright © OASIS Open 2011. 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 http://www.oasis-open.org/who/trademark.php for above guidance.

# Table of Contents

# 1 Introduction

[All text is normative unless otherwise labeled]

This document describes common components and component types that can be reused across the suite of Emergency Data Exchange Language (EDXL) standards. This document is intended for internal use by the Emergency Management Technical Committee and its subcommittees as they develop specific standards utilizing these types. The goal is to enable reuse of components which are commonly used in specifications and which have been designed based on lessons learned from the development of the Common Alert Protocol 1.1, the Distribution Element 1.0, Hospital Availability and Resource Messaging. The first use of these common components is intended to be in Situation Reports 1.0 and the Distribution Element 2.0. The components will be used and expanded as needed for future EDXL specifications.

## 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

**[RFC2119]**      S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

**[RFC2046]**      N. Freed, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, http://www.ietf.org/rfc/rfc2046.txt, IETF RFC 2046, November 1996.

**[RFC3066]**      H. Alvestrand, Tags for the Identification of Languages, http://www.ietf.org/rfc/rfc3066.txt, IETF RFC 3066, January 2001.

**[WGS 84]**       National Geospatial Intelligence Agency, Department of Defense World Geodetic System 1984, http://earth-info.nga.mil/GandG/tr8350_2.html, NGA Technical Report TR8350.2, January 2000.

**[XML 1.0]**      T. Bray, *Extensible Markup Language (XML) 1.0 (Third Edition)*, http://www.w3.org/TR/REC-xml/, W3C REC-XML-20040204, February 2004.

**[namespaces]**   T. Bray, *Namespaces in XML*, http://www.w3.org/TR/REC-xml-names/, W3C REC-xml-names-19990114, January 1999.

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

## 1.3 Non-Normative References

**[EDXL GFR]** *EDXL General Functional Requirements*, http://www.oasis-open.org/committees/document.php?document_id=10031&wg_abbrev=emergency, November 2004.

**[EDXL-DE IG]** *EDXL Distribution Element Implementer's Guide*, http://www.oasis-open.org/committees/document.php?document_id=14120&wg_abbrev=emergency, August 2005.

# 2    Design Principles & Concepts (non-normative)

## 2.1    Design Philosophy

Below are some of the guiding principles of the EDXL CT Specification:

1. Provide a method to capture and reuse xml types and elements for representing persons and organizations which are commonly needed across multiple EDXL standards.

2. Provide flexible mechanisms to update the EDXL CT Specification efficiently, without slowing down the EDXL standards development process.

3. Allow for easy updates to capture fixes or improvements.

4. Ease the reuse and understanding of the basic data components and protocols, important for emergency management and common across multiple specifications.

5. Speed the development of EDXL Standards through reuse of common components and thereby improve 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.

6. Support the integration of data elements from profiles which enables efficient and effective reuse of other important open standards.

## 2.2    Structural Summary

About multiplicity notation: "[l..h]" designates range from lower bound l to higher bound h, with both l and h Natural numbers, 0 ≤ l ≤ h, plus option "*" (unbounded) for h.

### 2.2.1    Simple Types

| Type name | depends on |
|---|---|
| – EDXLDateTimeType | xs:dateTime |
| – EDXLStringType | xs:token |
| – PercentageType | xs:float |
| – ValueListURIType | xs:anyURI |
| – ValueType | xs:string |
| – CurrencyType | xs:string |
| – DegreesCType | xs:float [-100 .. 70] |
| – DegreesCircleType | xs:float [0 .. 360] |
| – WeatherQualifierType | xs:string [*enumeration*] |
| – WeatherDescriptorType | xs:string [*enumeration*] |

| | | |
|---|---|---|
| – WeatherPrecipitationType | xs:string [*enumeration*] | |
| – WeatherObscurationType | xs:string [*enumeration*] | |
| – WeatherAddlPhenomType | xs:string [*enumeration*] | |
| – SkyConditionType | xs:string [*enumeration*] | |
| – RemarksType | xs:string | |
| – EstimateType | xs:boolean | |

## 2.2.2  Complex Types

| Type name | | depends on |
|---|---|---|
| – ValueListType | | ct:ValueListURI, ct:Value |
| – ValueKeyType | | ct:ValueListURI, ct:Value |
| – ValueKeyStringPairType | | ct:ValueKeyType, xs:string |
| – ValueKeyIntPairType | | ct:ValueKeyType, xs:int |
| – TimePeriodType | | ct:EDXLDateTimeType |
| – PersonTimePairType | | ct:PersonDetailsType, ct:EDXLDateTimeType |
| – OrganizationInformationType | | xpil:OrganizationDetailsType |
| – PersonDetailsType | | xpil:PersonDetailsType |
| – METARType | | *sequence of elements* |
| – StationID | [1,1] | xs:string *restricted* |
| – ObservationTime | [1,1] | ct:EDXLDateTimeType |
| – TempC | [0,1] | ct:DegreesCType |
| – DewPointC | [0,1] | ct:DegreesCType |
| – WindDirDegrees | [0,1] | ct:DegreesCircleType |
| – WindSpeedkt | [0,1] | xs:int [0 .. 300] |
| – WindGustkt | [0,1] | xs:int [0 .. 300] |
| – VisibilityStatuteMI | [0,1] | xs:float [0.0 .. 10.0] |
| – AltimeterHP | [0,1] | xs:int [800 .. 1200] |
| – SeaLevelPressuremb | [0,1] | xs:int [800 .. 1200] |
| – WeatherPhenomenaReport | [0,1] | *sequence of elements* |
| – Qualifier | [0,1] | ct:WeatherQualifierType |
| – Descriptor | [0,1] | ct:WeatherDescriptorType |
| – Precipitation | [0,1] | ct:WeatherPrecipitationType |
| – Obscuration | [0,1] | ct:WeatherObscurationType |
| – Additional | [0,1] | ct:WeatherAddlPhenomType |

| | | | |
|---|---|---|---|
| – | SkyCondition | [0,1] | ct:SkyConditionType |
| – | Precip1HrIn | [0,1] | xs:float *restricted* |
| – | Precip3HrIn | [0,1] | xs:float *restricted* |
| – | Precip6HrIn | [0,1] | xs:float *restricted* |
| – | Precip24HrIn | [0,1] | xs:float *restricted* |
| – | WeatherInfoType | | *sequence of elements* |
| – | METARString | [0,1] | xs:string |
| – | METARReadings | [0,1] | ct:METARType |
| – | WeatherRemarks | [0,1] | xs:string |
| – | WeatherConcerns | [0,1] | xs:string |
| – | EDXLGeoPoliticalLocationType | | *choice of elements* |
| – | Address | [1,1] | xal:AddressType |
| | **xor** | | |
| – | GeoCode | [1,1] | ct:ValueListType |
| – | EDXLLocationType | | *choice of elements* |
| – | EDXLGeoLocation | [1,1] | edxl-gsf:EDXLGeoLocationType |
| | **xor** | | |
| – | EDXLGeoPoliticalLocation | [1,1] | ct:EDXLGeoPoliticalLocationType |

## 2.2.3   Top Level Elements

| | Element name | | depends on |
|---|---|---|---|
| – | ValueListURI | | ct:ValueListURIType |
| – | Value | | ct:ValueType |
| – | WeatherInfo | | ct:WeatherInfoType |

# 3   EDXL Common Types Structure (normative)

## 3.1   Data Dictionary

Namespaces and prefixes used below include:

**xs="**`http://www.w3.org/2001/XMLSchema`**"**
**ct="**`urn:oasis:names:tc:emergency:edxl:ct:1.0`**"**
**xpil="**`urn:oasis:names:tc:emergency:edxl:ciq:1.0:xpil`**"**
**xal="**`urn:oasis:names:tc:emergency:edxl:ciq:1.0:xal`**"**
**xnl="**`urn:oasis:names:tc:emergency:edxl:ciq:1.0:xnl`**"**

### 3.1.1   EDXL Common Simple Types

| Type | EDXLDateTimeType |
|---|---|
| BaseType | Restricted `xs:dateTime` |
| Restriction | Pattern `"\d\d\d\d-\d\d-\d\dT\d\d:\d\d:\d\d[-,+]\d\d:\d\d"` |
| Usage | Use wherever you would otherwise use `xs:dateTime` |
| Definition | A restricted form of `dateTime` which requires the use of a timezone offset and thereby prohibits the use of "Z" without an offset. Also prohibited is the use fractional seconds. |
| Comments | 1.   The uniform requirement for a timezone offset provides greater reliability and robustness for emergency systems. |
| Schema Component | `<xs:simpleType name="EDXLDateTimeType">`<br>`  <xs:restriction base="xs:dateTime">`<br>`    <xs:pattern value="\d\d\d\d-\d\d-\d\dT\d\d:\d\d:\d\d[-,+]\d\d:\d\d"/>`<br>`  </xs:restriction>`<br>`</xs:simpleType>` |
| Used In | Top level type |
| Example | `<DateTimeSent>2009-11-15T16:53:00-05:00</DateTimeSent>` |

| Type | EDXLStringType |
|---|---|
| BaseType | Restricted `xs:token` |
| Restriction | `minLength = 1, maxLength = 1023` |
| Usage | Use wherever you would otherwise use xs:string |
| Definition | A restricted form of string which is limited to 1023 characters (and must be at least 1 character) in length |
| Comments | 1.   This common type provides a string type which is of long but limited length. Emer- |

| | |
|---|---|
| | gency systems shouldn't be required to manage indefinitely long string lengths. `maxLength` counts the maximum number of characters in the string.<br>2. This type does not exclude the use of the more general `xs:string`, but should be applied whenever length limitation is technically indicated, e.g.<br>- to prevent circumvention of REQUIRED usage by supplying an empty string (length 0), or<br>- for coding or transmission efficiency. |
| Schema Component | `<xs:simpleType name="EDXLStringType">`<br>  `<xs:restriction base="xs:token">`<br>    `<xs:maxLength value="1023"/>`<br>    `<xs:minLength value="1"/>`<br>  `</xs:restriction>`<br>`</xs:simpleType>` |
| Used In | Top level type |
| Example | `<SenderID>mary.thompson@myagency.gov</SenderID>` |

| Type | **PercentageType** |
|---|---|
| BaseType | Restricted `xs:float` |
| Restriction | `minInclusive=0, maxInclusive=100.0` |
| Usage | Use wherever you need to use a percentage |
| Definition | A restricted form of unsigned floating number ranging from 0.0 to 100.0 inclusive intended to represent a percentage |
| Comments | 1. Percentages are often used in emergency messages so this Percentage type facilitates a standardized format as opposed to ad hoc percentage formats. |
| Schema Component | `<xs:simpleType name="PercentageType">`<br>  `<xs:restriction base="xs:float">`<br>    `<xs:minInclusive value="0"/>`<br>    `<xs:maxInclusive value="100.0"/>`<br>  `</xs:restriction>`<br>`</xs:simpleType>` |
| Used In | Top level type |
| Example | `<Percentage>100<Percentage>` |

| Type | **ValueListURIType** |
|---|---|
| BaseType | Restricted `xs:anyURI` |
| Restriction | None. |
| Usage | Used to denote the URI of a `ValueList` and related types |
| Definition | A URI referencing an externally-managed list of values. |
| Comments | 1. A lesson learned from early EDXL specification development was the need to support lists of values that may vary depending on the jurisdiction or community. The `ValueListType` and related structures are based on the concept that the "list" of values can be maintained externally and referenced in the EDXL standards. The reference is handled by structures which include a `ValueListURI` providing a unique identifier for |

| | |
|---|---|
| | the external "list" and then followed by a value or values from that list. The reason "list" is quoted is because the external structure may be an ontology or other structure adopted by the jurisdiction or community rather than just a simple list. |
| Schema Component | `<xs:simpleType name="ValueListURIType">`<br>`  <xs:restriction base="xs:anyURI"/>`<br>`</xs:simpleType>` |
| Used In | Top level type |
| Examples | |

| Type | **ValueType** |
|---|---|
| BaseType | Restricted `xs:string` |
| Restriction | None. |
| Usage | Used to denote the value(s) of a `ValueList` and related types |
| Definition | A string value from an externally-managed list of values referenced by a `ValueListURI`. |
| Comments | 1. A lesson learned from early EDXL specification development was the need to support lists of values that may vary depending on the jurisdiction or community. The `ValueListType` and related structures are based on the concept that the "list" of values can be maintained externally and referenced in the EDXL standards. The reference is handled by structures which include a `ValueListURI` providing a unique identifier for the external "list" and then followed by a value or values from that "list". The reason "list" is quoted is because the external structure may be an ontology or other structure adopted by the jurisdiction or community rather than just a simple list. |
| Schema Component | `<xs:simpleType name="Value">`<br>`  <xs:restriction base="xs:string"/>`<br>`</xs:simpleType>` |
| Used In | Top level type |
| Examples | |

| Type | **RemarksType** |
|---|---|
| BaseType | Restricted `xs:string` |
| Restriction | None. |
| Usage | |
| Definition | General comments or remarks associated with any applicable element. |
| Comments | 1. Initially used in EDXL-SitRep SituationInformation, ManagementReportingSummary" and CasualtyAndIllnes Summary Report Types.<br>2. Source: ICS 201 |
| Schema Component | `<xs:simpleType name="RemarksType">`<br>`  <xs:restriction base="xs:string"/>`<br>`</xs:simpleType>` |
| Used In | Top level type |

| Examples | " Wildcat Canyon Mudslide"  Disaster declared by MyCounty was initially declared by MyTownship as "Pleasant Creek Neighborhood Sinkhole." Incident. |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------|

| Type | **EstimateType** |
|------|-----------------|
| BaseType | `xs:boolean` |
| Restriction | None. |
| Usage | |
| Definition | To designate whether a number or figure is actual or estimated.  Values include:<br>  'Y" = Estimated<br>  "N" = Not Estimated. |
| Comments |  Source: ICS 209 |
| Schema Component | `<xs:simpleType name="EstimateType">`<br>  `<xs:restriction base="xs:boolean"/>`<br>`</xs:simpleType>` |
| Used In | Top level type |
| Examples | |

| Type | **CurrencyType** |
|------|-----------------|
| BaseType | `xs:string` |
| Restriction | Pattern `"([0-9])+[.][0-9][0-9] [A-Z][A-Z][A-Z]"` |
| Usage | Use wherever currency is used in a specification. |
| Definition | A `CurrencyType` is at least one number followed by 0 or more numbers, followed by an optional fractional part, and followed by three capital letters designating the currency code (ISO 4217). |
| Comments | |
| Schema Component | `<xs:simpleType name="CurrencyType">`<br>  `<xs:restriction base="xs:string">`<br>    `<xs:pattern value="([0-9])+[.][0-9][0-9] [A-Z][A-Z][A-Z]"/>`<br>  `</xs:restriction>`<br>`</xs:simpleType>` |
| Used In | Top level type |
| Examples | `<ct:Currency>45USD</ct:Currency>`<br><br>`<ct:Currency`<br>`xsi:schemaLocation="urn:oasis:names:tc:emergency:edxl:ct:1.0: EDXL_Common_Types_ver05.xsd" xmlns:xsi="`http://www.w3.org/2001/XMLSchema-instance`"xmlns:ct="urn:oasis:names:tc:emergency:EDXL:CT:1.0">099999999999999999.00 AAA</ct:Currency>` |

| Type | **DegreesCType** |
| --- | --- |
| BaseType | Restricted `xs:float` |
| Restriction | `minInclusive=-100.0, maxInclusive=70.0` |
| Usage | Use wherever degree Celsius is used in a specification. |
| Definition | A restricted form of floating number ranging from -100.0 to 70.0 inclusive, intended to represent a temperature in degrees centigrade |
| Comments | |
| Schema Component | `<xs:simpleType name="DegreesCType">`<br>  `<xs:restriction base="xs:float">`<br>    `<xs:minInclusive value="-100.0"/>`<br>      `<xs:maxInclusive value="70.0"/>`<br>  `</xs:restriction>`<br>`</xs:simpleType>` |
| Used In | `METARType.TempC` |
| Examples | `<TempC>37.2</TempC>` |

| Type | **DegreesCircleType** |
| --- | --- |
| BaseType | Restricted `xs:float` |
| Restriction | `minInclusive= 0.0, maxInclusive=360.0` |
| Usage | Use wherever an angle measurement in degrees is used in a specification. |
| Definition | A restricted form of unsigned floating number ranging from 0.0 to 360.0 inclusive, intended to represent an angle measurement in degrees. |
| Comments | |
| Schema Component | `<xs:simpleType name="DegreesCircleType">`<br>  `<xs:restriction base="xs:float">`<br>    `<xs:minInclusive value="0.0"/>`<br>    `<xs:maxInclusive value="360.0"/>`<br>  `</xs:restriction>`<br>`</xs:simpleType>` |
| Used In | `METARType.WindDirDegrees` |
| Examples | `<WindDirDegrees>32.3</WindDirDegrees>` |

## 3.1.2  EDXL Enumerated Types

| Type | **WeatherQualifierType** |
| --- | --- |
| BaseType | `Enumeration` |
| Values | `"Light", "Moderate", "Heavy"` |
| Usage | |
| Definition | A selection of qualifiers to categorize types of weather. |
| Comments | |

| | |
|---|---|
| Schema Component | ```xml
<xs:simpleType name="WeatherQualifierType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Light"/>
    <xs:enumeration value="Moderate"/>
    <xs:enumeration value="Heavy"/>
  </xs:restriction>
</xs:simpleType>
``` |
| Used In | METARType.WeatherPhenomenaReport.Qualifier |
| Examples | `<Qualifier>Light</Qualifier>` |


| Type | **WeatherDescriptorType** |
|---|---|
| BaseType | Enumeration |
| Values | "Shallow", "Blowing", "Patches", "Showers", "Partial", "Drifting", "Thunderstorm", "Freezing" |
| Usage | |
| Definition | A selection of weather characteristics. |
| Comments | |
| Schema Component | ```xml
<xs:simpleType name="WeatherDescriptorType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Shallow"/>
    <xs:enumeration value="Blowing"/>
    <xs:enumeration value="Patches"/>
    <xs:enumeration value="Showers"/>
    <xs:enumeration value="Partial"/>
    <xs:enumeration value="Drifting"/>
    <xs:enumeration value="Thunderstorm"/>
    <xs:enumeration value="Freezing"/>
  </xs:restriction>
</xs:simpleType>
``` |
| Used In | METARType.WeatherPhenomenaReport.Descriptor |
| Examples | `<Descriptor>Showers</Descriptor>` |


| Type | **WeatherPrecipitationType** |
|---|---|
| BaseType | Enumeration |
| Values | "Drizzle", "Ice Crystals", "Unknown", "Rain", "Ice Pellets", "Snow", "Hail", "Snow Grains", "Snow Hail" |
| Usage | |
| Definition | A selection of precipitation characteristics. |
| Comments | |
| Schema Component | ```xml
<xs:simpleType name="WeatherPrecipitationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Drizzle"/>
    <xs:enumeration value="Ice Crystals"/>
    <xs:enumeration value="Unknown"/>
    <xs:enumeration value="Rain"/>
``` |

```
            <xs:enumeration value="Ice Pellets"/>
            <xs:enumeration value="Snow"/>
            <xs:enumeration value="Hail"/>
            <xs:enumeration value="Snow Grains"/>
            <xs:enumeration value="Snow Hail"/>
        </xs:restriction>
    </xs:simpleType>
```

| | |
|---|---|
| Used In | METARType.WeatherPhenomenaReport.Precipitation |
| Examples | `<Precipitation>Drizzle</Precipitation>` |


| Type | **WeatherObscurationType** |
|---|---|
| BaseType | Enumeration |
| Values | "Mist", "Sand", "Smoke", "Haze", "Volcanic Ash", "Spray", "Widespread Dust", "Other" |
| Usage | |
| Definition | A selection of qualifiers to categorize types of obscuration. |
| Comments | |
| Schema Component | `<xs:simpleType name="WeatherObscurationType">`<br>`  <xs:restriction base="xs:string">`<br>`    <xs:enumeration value="Mist"/>`<br>`    <xs:enumeration value="Sand"/>`<br>`    <xs:enumeration value="Smoke"/>`<br>`    <xs:enumeration value="Haze"/>`<br>`    <xs:enumeration value="Volcanic Ash"/>`<br>`    <xs:enumeration value="Spray"/>`<br>`    <xs:enumeration value="Widespread Dust"/>`<br>`    <xs:enumeration value="Other"/>`<br>`  </xs:restriction>`<br>`</xs:simpleType>` |
| Used In | METARType.WeatherPhenomenaReport.Obscuration |
| Examples | `<Obscuration>Other</Obscuration>` |


| Type | **WeatherAddlPhenomType** |
|---|---|
| BaseType | Enumeration |
| Values | "Squall", "Funnel Cloud", "Sandstorm", "Tornado", "Waterspout", "Dust-storm", "Dust Whirls" |
| Usage | |
| Definition | A selection of qualifiers for weather phenomena. |
| Comments | |
| Schema Component | `<xs:simpleType name="WeatherAddlPhenomType">`<br>`  <xs:restriction base="xs:string">`<br>`    <xs:enumeration value="Squall"/>`<br>`    <xs:enumeration value="Funnel Cloud"/>`<br>`    <xs:enumeration value="Sandstorm"/>`<br>`    <xs:enumeration value="Tornado"/>` |

| | |
|---|---|
| | ```
        <xs:enumeration value="Waterspout"/>
        <xs:enumeration value="Duststorm"/>
        <xs:enumeration value="Dust Whirls"/>
    </xs:restriction>
</xs:simpleType>
``` |
| Used In | METARType.WeatherPhenomenaReport.Additional |
| Examples | `<Additional>Dust Whirls</Additional>` |

<br>

| Type | **SkyConditionType** |
|---|---|
| BaseType | Enumeration |
| Values | "Sky Clear", "Few", "Scattered", "Broken", "Overcast" |
| Usage | |
| Definition | A selection of qualifiers for sky conditions. |
| Comments | |
| Schema Component | ```
<xs:simpleType name="SkyConditionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Sky Clear"/>
    <xs:enumeration value="Few"/>
    <xs:enumeration value="Scattered"/>
    <xs:enumeration value="Broken"/>
    <xs:enumeration value="Overcast"/>
  </xs:restriction>
</xs:simpleType>
``` |
| Used In | METARType.SkyCondition |
| Examples | `<SkyCondition>Overcast</SkyCondition>` |

### 3.1.3   EDXL Common Complex Types

| Type | **ValueListType** |
|---|---|
| BaseType | xs:complexType |
| Restriction | None. |
| Usage | Use wherever a specification needs values from an externally managed list. |
| Definition | A `ValueListType` includes one `ValueListURI` element and one or more `Value` elements. |
| Comments | 1.   A lesson learned from early EDXL specification development was the need to support lists of values that may vary depending on the jurisdiction or community. The `ValueListType` and related structures are based on the concept that the "list" of values can be maintained externally and referenced in the EDXL standards. The reference is handled by structures which include a `ValueListURI` providing a unique identifier for the external "list" and then followed by a value or values from that "list".  The reason "list" is quoted is because the external structure may be an ontology or other structure |

| | |
|---|---|
| | adopted by the jurisdiction or community rather than just a simple list.<br>2. A lesson learned is that enumerated types provide a brittle, hard-to-change solution to a list of types which needs to satisfy the needs of many jurisdictions. |
| Schema Component | ```xml<br><xs:complexType name="ValueListType"><br>  <xs:sequence><br>    <xs:element ref="ct:ValueListURI" minOccurs="1" maxOccurs="1"/><br>    <xs:element ref="ct:Value" minOccurs="1" maxOccurs="unbounded"/><br>  </xs:sequence><br></xs:complexType><br>``` |
| Used In | Top level type |
| Examples | ```xml<br><Keyword><br>  <ct:ValueListURI>urn:myagency:gov:sensors:keywords</ct:ValueListURI><br>  <ct:Value>SNM Detection</ct:Value><br>  <ct:Value>XYZ</ct:Value><br></Keyword><br>``` |


| Type | **ValueKeyType** |
|---|---|
| BaseType | `xs:complexType` |
| Restriction | None. |
| Usage | Use wherever a specification needs one single value from an externally managed list. |
| Definition | A `ValueKeyType` includes one `ValueListURI` element and one and only one `Value` element. |
| Comments | 1. A lesson learned from early EDXL specification development was the need to support lists of values that may vary depending on the jurisdiction or community. The `ValueKeyType` and related structures are based on the concept that the "list" of values can be maintained externally and referenced in the EDXL standards. The reference is handled by structures which include a `ValueListURI` providing a unique identifier for the external "list" and then followed by a value from that "list". The reason "list" is quoted is because the external structure may be an ontology or other structure adopted by the jurisdiction or community rather than just a simple list.<br>2. A lesson learned is that enumerated types provide a brittle, hard-to-change solution to a list of types which needs to satisfy the needs of many jurisdictions.<br>3. A lesson learned is that from some kinds of lists only one value is appropriate and multiple values would be an error. In this case, use `ValueKeyType` instead of `ValueListType`. |
| Schema Component | ```xml<br><xs:complexType name="ValueKeyType"><br>  <xs:sequence><br>    <xs:element ref="ct:ValueListURI" minOccurs="1" maxOccurs="1"/><br>    <xs:element ref="ct:Value" minOccurs="1" maxOccurs="1"/><br>  </xs:sequence><br></xs:complexType><br>``` |
| Used In | Top level type |
| Examples | ```xml<br><DistributionDefType><br>  <ct:ValueListURI><br>    urn:oasis:names:tc:emergency:EDXL:DE:2.0:Defaults:DistributionType<br>  </ct:ValueListURI><br>  <ct:Value>Report</ct:Value><br></DistributionDefType><br>``` |

| Type | **ValueKeyStringPairType** |
|---|---|
| BaseType | `xs:complexType` |
| Usage | Use wherever a specification needs one single value from an externally managed list paired with a string. |
| Definition | A `ValueKeyStringPairType` includes one `ValueKeyURI` (of type `ValueKeyType` containing a `ValueListURI` and one single `Value`) followed by a `PairValue` of type `xs:string`. |
| Comments | |
| Schema Component | ```
<xs:complexType name="ValueKeyStringPairType">
  <xs:sequence>
    <xs:element name="ValueKeyURI" type="ct:ValueKeyType"
        minOccurs="1" maxOccurs="1"/>
    <xs:element name="PairValue" type="xs:string"
        minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
``` |
| Used In | Top level type |
| Examples | ```
<AValueKeyStringPair>
  <ct:ValueKeyURI>
    <ct:ValueListURI>http://example.com/lists/mylist</ct:ValueListURI>
    <ct:Value>OneSingleValue</ct:Value>
  </ct:ValueKeyURI>
  <ct:PairValue>A Paired String</ct:PairValue>
</AValueKeyStringPair>
``` |


| Type | **ValueKeyIntPairType** |
|---|---|
| BaseType | `xs:complexType` |
| Usage | Use wherever a specification needs one single value from an externally managed list paired with an integer. |
| Definition | A `ValueKeyIntPairType` includes one `ValueKeyURI` (of type `ValueKeyType` containing a `ValueListURI` and one single `Value`) followed by a `PairValue` of type `xs:int`. |
| Comments | |
| Schema Component | ```
<xs:complexType name="ValueKeyIntPairType">
  <xs:sequence>
    <xs:element name="ValueKeyURI" type="ct:ValueKeyType"
        minOccurs="1" maxOccurs="1"/>
    <xs:element name="PairValue" type="xs:int"
        minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
``` |
| Used In | Top level type |
| Examples | ```
<AValueKeyIntPair>
  <ct:ValueKeyURI>
    <ct:ValueListURI>
        http://example.com/lists/mylist
    </ct:ValueListURI>
    <ct:Value>OneSingleValue</ct:Value>
  </ct:ValueKeyURI>
  <ct:PairValue>37</ct:PairValue>
``` |

| | |
|---|---|
| | `</AValueKeyIntPair>` |

| Type | **TimePeriodType** |
|---|---|
| BaseType | `xs:complexType` |
| Usage | Use wherever a specification needs to represent a period of time. |
| Definition | A `TimePeriodType` includes one and only one `FromDateTime` of type `ct:EDXLDateTimeType` and one and only one `ToDateTime` element of type `ct:EDXLDateTimeType` . |
| Comments | 1. Time periods are commonly needed in emergency standards. This type provides a simple and useful representation of a time period which can be used for uniformity throughout the EDXL specifications. |
| Schema Component | `<xs:complexType name="TimePeriodType">`<br>`  <xs:sequence>`<br>`    <xs:element name="FromDateTime" type="ct:EDXLDateTimeType"`<br>`        minOccurs="1" maxOccurs="1"/>`<br>`    <xs:element name="ToDateTime" type="ct:EDXLDateTimeType"`<br>`        minOccurs="1" maxOccurs="1"/>`<br>`  </xs:sequence>`<br>`</xs:complexType>` |
| Used In | Top level element |
| Examples | `<ATimePeriod>`<br>`  <ct:FromDateTime>2009-11-15T17:53:00-05:00</ct:FromDateTime>`<br>`  <ct:ToDateTime>2009-11-15T16:53:00-05:00</ct:ToDateTime>`<br>`</ATimePeriod>` |

| Type | **PersonTimePairType** |
|---|---|
| BaseType | `xs:complexType` |
| Usage | Use wherever a specification needs to represent a person paired with a time. |
| Definition | A `PersonTimePairType` includes one and only one `PersonDetails` element of type `ct:PersonDetailsType` and one and only one `TimeValue` of type `ct:EDXLDateTimeType`. |
| Comments | |
| Schema Component | `<xs:complexType name="PersonTimePairType">`<br>`  <xs:sequence>`<br>`    <xs:element name="PersonDetails" type="ct:PersonDetailsType"`<br>`        minOccurs="1" maxOccurs="1"/>`<br>`    <xs:element name="TimeValue" type="ct:EDXLDateTimeType"`<br>`        minOccurs="1" maxOccurs="1"/>`<br>`  </xs:sequence>`<br>`</xs:complexType>` |
| Used In | Top level type |
| Examples | `<APersonTimePair>`<br>`  <ct:PersonDetails>`<br>`    <xnl:PersonName>`<br>`      <xnl:NameElement>Mary Smith</n:NameElement>`<br>`    </xnl:PersonName>`<br>`  </ct:PersonDetails>`<br>`  <ct:TimeValue>2009-11-15T17:53:00-05:00</ct:TimeValue>` |

| | `</APersonTimePair>` |
|---|---|

| Type | **METARType** |
|---|---|
| BaseType | `xs:complexType` |
| Usage | |
| Definition | A subset of the METAR weather data set. |
| Comments | This is a verbose form for reporting METAR weather information |
| Sub-ele-ments | • `StationID` [1..1] of type `xs:string` **restricted**<br>• `ObservationTime` [1..1] of type `ct:EDXLDateTimeType`<br>• `TempC` [0..1] of type `ct:DegreesCType`<br>• `DewPointC` [0..1] of type `ct:DegreesCType`<br>• `WindDirDegrees` [0..1] of type `ct:DegreesCircleType`<br>• `WindSpeedkt` [0..1] of type `xs:int` **restricted**<br>• `WindGustkt` [0..1] of type `xs:int` **restricted**<br>• `VisibilityStatuteMI` [0..1] of type `xs:float` **restricted**<br>• `AltimeterHP` [0..1] of type `xs:int` **restricted**<br>• `SeaLevelPressuremb` [0..1] of type `xs:float` **restricted**<br>• `WeatherPhenomenaReport` [0..1] of type `xs:complexType`<br>• `SkyCondition` [0..1] of type `ct:SkyConditionType`<br>• `Precip1HrIn` [0..1] of type `xs:float` **restricted**<br>• `Precip3HrIn` [0..1] of type `xs:float` **restricted**<br>• `Precip6HrIn` [0..1] of type `xs:float` **restricted**<br>• `Precip24HrIn` [0..1] of type `xs:float` **restricted** |
| Schema Component | ```
See schema
<xs:complexType name="METARType">
  ..
</xs:complexType>
``` |
| Used In | Top level type |
| Examples | ```
<myMETAR>
  <ct:StationID>KEYF</ct:StationID>
  <ct:ObservationTime>2011-04-23T01:41:00+00:00</ct:ObservationTime>
  <ct:TempC>37.2</ct:TempC>
  <ct:DewpointC>10.0</ct:DewpointC>
  <ct:WindDirDegrees>32.3</ct:WindDirDegrees>
  <ct:WindSpeedkt>20</ct:WindSpeedkt>
  <ct:WindGustkt>50</ct:WindGustkt>
  <ct:VisibilityStatuteMI>1.0</ct:VisibilityStatuteMI>
  <ct:AltimeterHP>800</ct:AltimeterHP>
  <ct:SeaLevelPressuremb>800</ct:SeaLevelPressuremb>
  <ct:WeatherPhenomenaReport>
    ...
  </ct:WeatherPhenomenaReport>
  <ct:SkyCondition>Overcast</ct:SkyCondition>
  <ct:Precip1HrIn>00.01</ct:Precip1HrIn>
  <ct:Precip3HrIn>01.00</ct:Precip3HrIn>
  <ct:Precip6HrIn>01.23</ct:Precip6HrIn>
  <ct:Precip24HrIn>02.25</ct:Precip24HrIn>
</myMETAR>
``` |

| Sub-Element | [METARType.]**StationID** |
|---|---|
| Type | `xs:string` restricted |
| Restriction | Pattern "[A-Z]{4}" |
| Usage | [1..1] |
| Definition | Identifies the reporting station |
| Comments | Four-character ICAO Location Indicator |
| Schema Component | `<xs:element name="StationID" minOccurs="1">`<br>`  <xs:simpleType>`<br>`    <xs:restriction base="xs:string">`<br>`      <xs:pattern value="[A-Z]{4}"/>`<br>`    </xs:restriction>`<br>`  </xs:simpleType>`<br>`</xs:element>` |
| Used In | `METARType` |
| Examples | `<ct:StationID>KEYF</ct:StationID>` |


| Sub-Element | [METARType.]**WindSpeedkt** |
|---|---|
| Type | `xs:int` restricted |
| Restriction | Range [0 .. 300] |
| Usage | [0..1] |
| Definition | Wind speed in knots |
| Comments | |
| Schema Component | `<xs:element name="WindSpeedkt" minOccurs="0">`<br>`  <xs:simpleType>`<br>`    <xs:restriction base="xs:int">`<br>`      <xs:minInclusive value="0"/>`<br>`      <xs:maxInclusive value="300"/>`<br>`    </xs:restriction>`<br>`  </xs:simpleType>`<br>`</xs:element>` |
| Used In | `METARType` |
| Examples | `<ct:WindSpeedkt>20</ct:WindSpeedkt>` |


| Sub-Element | [METARType.]**WindGustkt** |
|---|---|
| Type | `xs:int` restricted |
| Restriction | Range [0 .. 300] |
| Usage | [0..1] |

| Definition | Speed of wind gusts in knots |
|---|---|
| Comments | |
| Schema Component | <pre><xs:element name="WindGustkt" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:int">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="300"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element></pre> |
| Used In | METARType |
| Examples | <ct:WindGustkt>50</ct:WindGustkt> |

| Sub-Ele-ment | [METARType.]**VisibilityStatuteMI** |
|---|---|
| Type | xs:float restricted |
| Restriction | Range [0 .. 10.0] |
| Usage | [0..1] |
| Definition | Visibility in Statute Miles |
| Comments | |
| Schema Component | <pre><xs:element name="VisibilityStatuteMI" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:float">
      <xs:minInclusive value="0.0"/>
      <xs:maxInclusive value="10.0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element></pre> |
| Used In | METARType |
| Examples | <ct:VisibilityStatuteMI>1.0</ct:VisibilityStatuteMI> |

| Sub-Ele-ment | [METARType.]**AltimeterHP** |
|---|---|
| Type | xs:int restricted |
| Restriction | Range [800 .. 1200] |
| Usage | [0..1] |
| Definition | Altimeter measurement in hectopascal |
| Comments | |
| Schema Component | <pre><xs:element name="AltimeterHP" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:int">
      <xs:minInclusive value="800"/>
      <xs:maxInclusive value="1200"/></pre> |

| | |
|---|---|
| | ```
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
``` |
| Used In | `METARType` |
| Examples | `<ct:AltimeterHP>800</ct:AltimeterHP>` |


| **Sub-Ele-ment** | `[METARType.]`**`SeaLevelPressuremb`** |
|---|---|
| Type | `xs:int` restricted |
| Restriction | Range [800 .. 1200] |
| Usage | [0..1] |
| Definition | Atmospheric pressure at sea level in millibar |
| Comments | 1 mb = 1 hPa |
| Schema Component | ```
<xs:element name="SeaLevelPressuremb" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:int">
      <xs:minInclusive value="800"/>
      <xs:maxInclusive value="1200"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
``` |
| Used In | `METARType` |
| Examples | `<ct:SeaLevelPressuremb>800</ct:SeaLevelPressuremb>` |


| **Sub-Ele-ment** | `[METARType.]`**`WeatherPhenomenaReport`** |
|---|---|
| Type | `xs:complexType` |
| Usage | [0..1] |
| Definition | |
| Comments | |
| Sub-ele-ments | <ul><li>`Qualifier` [0..1] of type `ct:WeatherQualifierType`</li><li>`Descriptor` [0..1] of type `ct:WeatherDescriptorType`</li><li>`Precipitation` [0..1] of type `ct:WeatherPrecipitationType`</li><li>`Obscuration` [0..1] of type `ct:WeatherObscurationType`</li><li>`Additional` [0..1] of type `ct:WeatherAddlPhenomType`</li></ul> |
| Schema Component | ```
<xs:element name="SeaLevelPressuremb" minOccurs="0">
  <xs:simpleType>
    <xs:restriction base="xs:int">
      <xs:minInclusive value="800"/>
      <xs:maxInclusive value="1200"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
``` |

| Used In | METARType |
|---|---|
| Examples | `<ct:WeatherPhenomenaReport>`<br>`  <ct:Qualifier>Light</ct:Qualifier>`<br>`  <ct:Descriptor>Showers</ct:Descriptor>`<br>`  <ct:Precipitation>Drizzle</ct:Precipitation>`<br>`  <ct:Obscuration>Other</ct:Obscuration>`<br>`  <ct:Additional>Dust Whirls</ct:Additional>`<br>`</ct:WeatherPhenomenaReport>` |


| Sub-Element | [METARType.]**Precip1HrIn** |
|---|---|
| Type | `xs:float` restricted |
| Restriction | Pattern "[0-9][0-9].[0-9][0-9]" |
| Usage | [0..1] |
| Definition | Amount of rain fall in 1 h, in inches |
| Comments | |
| Schema Component | `<xs:element name="Precip1HrIn" minOccurs="0">`<br>`  <xs:simpleType>`<br>`    <xs:restriction base="xs:float">`<br>`      <xs:pattern value="[0-9][0-9].[0-9][0-9]"/>`<br>`    </xs:restriction>`<br>`  </xs:simpleType>`<br>`</xs:element>` |
| Used In | METARType |
| Examples | `<ct:Precip1HrIn>00.01</ct:Precip1HrIn>` |


| Sub-Element | [METARType.]**Precip3HrIn** |
|---|---|
| Type | `xs:float` restricted |
| Restriction | Pattern "[0-9][0-9].[0-9][0-9]" |
| Usage | [0..1] |
| Definition | Amount of rain fall in 3 h, in inches |
| Comments | |
| Schema Component | `<xs:element name="Precip3HrIn" minOccurs="0">`<br>`  <xs:simpleType>`<br>`    <xs:restriction base="xs:float">`<br>`      <xs:pattern value="[0-9][0-9].[0-9][0-9]"/>`<br>`    </xs:restriction>`<br>`  </xs:simpleType>`<br>`</xs:element>` |
| Used In | METARType |
| Examples | `<ct:Precip3HrIn>01.00</ct:Precip3HrIn>` |

| Sub-Element | [METARType.]**Precip6HrIn** |
|---|---|
| Type | `xs:float` restricted |
| Restriction | Pattern "[0-9][0-9].[0-9][0-9]" |
| Usage | [0..1] |
| Definition | Amount of rain fall in 6 h, in inches |
| Comments | |
| Schema Component | `<xs:element name="Precip6HrIn" minOccurs="0">`<br>`  <xs:simpleType>`<br>`    <xs:restriction base="xs:float">`<br>`      <xs:pattern value="[0-9][0-9].[0-9][0-9]"/>`<br>`    </xs:restriction>`<br>`  </xs:simpleType>`<br>`</xs:element>` |
| Used In | `METARType` |
| Examples | `<ct:Precip6HrIn>01.23</ct:Precip6HrIn>` |

| Sub-Element | [METARType.]**Precip24HrIn** |
|---|---|
| Type | `xs:float` restricted |
| Restriction | Pattern "[0-9][0-9].[0-9][0-9]" |
| Usage | [0..1] |
| Definition | Amount of rain fall in 24 h, in inches |
| Comments | |
| Schema Component | `<xs:element name="Precip24HrIn" minOccurs="0">`<br>`  <xs:simpleType>`<br>`    <xs:restriction base="xs:float">`<br>`      <xs:pattern value="[0-9][0-9].[0-9][0-9]"/>`<br>`    </xs:restriction>`<br>`  </xs:simpleType>`<br>`</xs:element>` |
| Used In | `METARType` |
| Examples | `<ct:Precip24HrIn>02.25</ct:Precip24HrIn>` |

| Type | **WeatherInfoType** |
|---|---|
| BaseType | `xs:complexType` |
| Usage | Use wherever weather info is needed in a specification. |
| Definition | A container to transmit predefined weather info with free format remarks and concerns |
| Comments | 1. METAR string: raw METAR data, "the most popular format in the world for the transmission of weather data. It is highly standardized through International Civil Aviation Organization (ICAO), which allows it to be understood throughout most of the world" |

| | |
|---|---|
| | [Wikipedia]<br>2.   METAR readings: a more verbose formatted set of weather data |
| Sub-ele-ments | •   METARString [0..1] of type xs:string<br>•   METARReadings [0..1] of type ct:METARType<br>•   WeatherRemarks [0..1] of type xs:string<br>•   WeatherConcerns [0..1] of type xs:string |
| Schema Component | <pre><xs:complexType name="WeatherInfoType"><br>  <xs:sequence><br>    <xs:element name="METARString" type="xs:string" minOccurs="0"/><br>    <xs:element name="METARReadings" type="ct:METARType" minOccurs="0"/><br>    <xs:element name="WeatherRemarks" type="xs:string" minOccurs="0"/><br>    <xs:element name="WeatherConcerns" type="xs:string" minOccurs="0"/><br>  </xs:sequence><br></xs:complexType></pre> |
| Used In | Top level type |
| Examples | <pre><ct:WeatherInfo<br>xsi:schemaLocation="urn:oasis:names:tc:emergency:edxl:ct:1.0<br>EDXL_Common_Types_wd02_dpm.xsd" xmlns:xsi="http://www.w3.org/2001/XMLS-<br>chema-instance" xmlns:ct="urn:oasis:names:tc:emergency:edxl:ct:1.0"><br>  <ct:METARString>KEYF 222355Z AUTO 00000KT 4SM BR 17/17 A3022 RMK AO2<br>T01700170</ct:METARString><br>  <ct:METARReadings><br>    <ct:StationID>KEYF</ct:StationID><br>    <ct:ObservationTime>2011-04-23T01:41:00+00:00</ct:ObservationTime><br>    <ct:TempC>37.2</ct:TempC><br>    <ct:DewpointC>10.0</ct:DewpointC><br>    <ct:WindDirDegrees>32.3</ct:WindDirDegrees><br>    <ct:WindSpeedkt>20</ct:WindSpeedkt><br>    <ct:WindGustkt>50</ct:WindGustkt><br>    <ct:VisibilityStatuteMI>1.0</ct:VisibilityStatuteMI><br>    <ct:AltimeterHP>800</ct:AltimeterHP><br>    <ct:SeaLevelPressuremb>800</ct:SeaLevelPressuremb><br>    <ct:WeatherPhenomenaReport><br>      <ct:Qualifier>Light</ct:Qualifier><br>      <ct:Descriptor>Showers</ct:Descriptor><br>      <ct:Precipitation>Drizzle</ct:Precipitation><br>      <ct:Obscuration>Other</ct:Obscuration><br>      <ct:Additional>Dust Whirls</ct:Additional><br>    </ct:WeatherPhenomenaReport><br>    <ct:SkyCondition>Overcast</ct:SkyCondition><br>    <ct:Precip1HrIn>00.01</ct:Precip1HrIn><br>    <ct:Precip3HrIn>01.00</ct:Precip3HrIn><br>    <ct:Precip6HrIn>01.23</ct:Precip6HrIn><br>    <ct:Precip24HrIn>02.25</ct:Precip24HrIn><br>  </ct:METARReadings><br>  <ct:WeatherRemarks>This is weather</ct:WeatherRemarks><br>  <ct:WeatherConcerns><br>    I am concerned it may change, and that scares me...<br>  </ct:WeatherConcerns><br></ct:WeatherInfo></pre> |

| Type | **OrganizationInformationType** |
|---|---|
| BaseType | Extends xpil:OrganisationDetailsType |
| Usage | Use wherever a specification needs to specify information about an organization. |
| Definition | The container type for organization information elements. The OrganizationInforma- |

| | |
|---|---|
| | `tionType` includes at least one `xnl:OrganisationName` and optionally Addresses, `Con-tactNumbers`, `ElectronicAddressIdentifiers` and `OrganisationInfo`. See the OASIS EM CIQ Profile for details. |
| Comments | 1. Note that some elements use the American spelling "Organization" and some the English spelling "Organisation". |
| Schema Component | `<xs:complexType name="OrganizationInformationType">`<br>  `<xs:complexContent>`<br>    `<xs:extension base="xpil:OrganisationDetailsType"/>`<br>  `</xs:complexContent>`<br>`</xs:complexType>` |
| Used In | Top level type |
| Examples | `<AnOrganizationInformation>`<br>  `<xnl:OrganisationName>`<br>    `<xnl:NameElement>Corporation XYZ</xnl:NameElement>`<br>  `</xnl:OrganisationName>`<br>`</AnOrganizationInformation>` |


| Type | **`PersonDetailsType`** |
|---|---|
| Type | `xpil:PersonDetailsType` |
| Usage | Used in the `PersonTimePairType`. |
| Definition | `A container for defining the unique characteristics of a person only.` `PersonDetailsType` is an extension of `xpil:PersonDetailsType` which is defined in the OASIS EM TC CIQ profile xpil schema to include at least one PersonName, and optionally one `Addresses`, `ContactNumbers`, `ElectronicAddressIdentifiers` and `Identifers`. For more information, see the OASIS EM TC CIQ profile. |
| Comments | 1. See the EM-TC CIQ Profile |
| Schema Component | `<xs:complexType name="PersonDetailsType">`<br>  `<xs:complexContent>`<br>    `<xs:extension base="xpil:PersonDetailsType"/>`<br>  `</xs:complexContent>`<br>`</xs:complexType>` |
| Used In | `PersonTimePairType` |
| Examples | `<APersonDetails>`<br>  `<ct:PersonDetails>`<br>    `<xnl:PersonName>`<br>      `<xnl:NameElement>Mary Smith</n:NameElement>`<br>    `</xnl:PersonName>`<br>  `</ct:PersonDetails>`<br>`</APersonDetails>` |


| Type | **`EDXLGeoPoliticalLocationType`** |
|---|---|
| BaseType | `xs:complexType` |
| Restriction | Choice. |
| Usage | Use wherever a specification needs a geopolitical location described as address or by geo-code. |

| Definition | A container for defining Geo-Political Locations. |
|---|---|
| Comments | |
| Schema Component | <xs:complexType name="EDXLGeoPoliticalLocationType">  <xs:choice>    <xs:element name="Address" type="xal:AddressType"/>    <xs:element name="GeoCode" type="ct:ValueListType"/>  </xs:choice> </xs:complexType> |
| Used In | Top level type |
| Examples | |

| Type | **EDXLLocationType** |
|---|---|
| BaseType | xs:complexType |
| Restriction | Choice. |
| Usage | Use wherever a specification needs a designation of a location. |
| Definition | A Container for describing both Geo-Political and Geographic Locations. |
| Comments | |
| Schema Component | <xs:complexType name="EDXLLocationType">  <xs:choice>    <xs:element name="EDXLGeoLocation"            type="edxl-gsf:EDXLGeoLocationType"/>    <xs:element name="EDXLGeoPoliticalLocation"            type="ct:EDXLGeoPoliticalLocationType"/>  </xs:choice> </xs:complexType> |
| Used In | Top level type |
| Examples | |

## 3.1.4   EDXL Common Top Level Elements

| Element | **ValueListURI** |
|---|---|
| Type | ct:ValueListURIType |
| Usage | Used to denote the URI of a ValueListType and related types |
| Definition | A  URI referencing an externally-managed list of values. |
| Comments | 1.  A lesson learned from early EDXL specification development was the need to support lists of values that may vary depending on the jurisdiction or community. The ValueListType and related structures are based on the concept that the "list" of values can be maintained externally and referenced in the EDXL standards. The reference is handled by structures which include a ValueListURI providing a unique identifier for the external "list" and then followed by a value or values from that list. The reason "list" is quoted is because the external structure may be an ontology or other structure adopted by the jurisdiction or community rather than just a simple list. |

| Schema Component | `<xs:element name="ValueListURI" type="ValueListURIType"/>` |
|---|---|
| Used In | `ValueListType`<br>`ValueKeyType`<br>`ValueKeyStringPairType`<br>`ValueKeyIntPairType` |
| Examples | `<ct:ValueListURI>`http://example.com/mylist`</ct:ValueListURI>`<br><br>`<ct:ValueListURI>`<br>  `urn:oasis:names:tc:emergency:edxl:de:2.0:Defaults:DistributionType`<br>`</ct:ValueListURI>` |

| **Element** | **Value** |
|---|---|
| Type | `ct:ValueType` |
| Usage | Used to denote the value(s) of a `ValueListType` and related types |
| Definition | A string value from an externally-managed list of values referenced by a `ValueListURI`. |
| Comments | 1. A lesson learned from early EDXL specification development was the need to support lists of values that may vary depending on the jurisdiction or community. The `ValueListType` and related structures are based on the concept that the "list" of values can be maintained externally and referenced in the EDXL standards. The reference is handled by structures which include a `ValueListURI` providing a unique identifier for the external "list" and then followed by a value or values from that "list". The reason "list" is quoted is because the external structure may be an ontology or other structure adopted by the jurisdiction or community rather than just a simple list. |
| Schema Component | `<xs:element name="Value" type="ValueType"/>` |
| Used In | `ValueListType`<br>`ValueKeyType`<br>`ValueKeyStringPairType`<br>`ValueKeyIntPairType` |
| Examples | `<ct:Value>SomeValue</ct:Value>` |

# 4   Conformance

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

TBD

# Appendix A  Acknowledgements

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

**Participants:**

Don McGarry, MITRE Corp., Member
Jeff Waters, DoD, Member
Rex Brooks, Network Centric Operations Industry Consortium (NCOIC), Member
Werner Joerg, IEM, Inc., Member

# Appendix B  Non-Normative Text

# Appendix C  Revision History

| Revision | Date | Editor | Changes Made |
|---|---|---|---|
| WD01 | 03/02/2011 | Jeff Waters | Initial Setup |
| WD02 | 04/21/2011 | Werner Joerg | Adaptation to new schema; ready for TC review |
| WD03 | 05/02/2011 | Werner Joerg | Expanded WeatherInfo; ready for TC review |
|  | 05/10/2011 | Werner Joerg | Fixed link for [WGS 84] reference |
| WD04 | 11/15/2011 | Werner Joerg | Added RemarksType, EstimateType, EDXLGeoPoliticalLocationType and EDXLLocationType; fixed statements in 2.1 |