

Addressing — Machine interchange syntax

WD/CD/DIS/FDIS stage

Warning for WDs and CDs

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

To help you, this guide on writing standards was produced by the ISO/TMB and is available at <https://www.iso.org/iso/how-to-write-standards.pdf>

A model manuscript of a draft International Standard (known as “The Rice Model”) is available at https://www.iso.org/iso/model_document-rice_model.pdf

© ISO 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

2.1 General.....	1
2.2 AddressProfileDescription.....	1
2.3 AddressInstance.....	1
2.4 AddressInputTemplate	1
2.5 AddressDisplayTemplate	1
5.1 Address Profiles.....	5
5.1.1 Creating Address Profiles	5
5.1.2 Publishing Address Profiles	5
5.1.3 Updating Address Profiles	5
5.1.4 Using Address Profiles.....	5
5.1.5 Retiring Address Profiles	5
5.2 Address Instances	6
5.2.1 Creating an Address Instance	6
5.2.2 Sending an Address Instance.....	6
5.2.3 Displaying an Address Instance.....	6
5.2.4 Improving Quality of an Address Instance	6
5.2.5 Verifying an Address Instance.....	6
5.2.6 Adding Associated Data to an Address Instance	6
5.2.7 Discarding an Address Instance.....	6
6.1 General Structure.....	7
6.2 Address Profile Definition (AddressProfileDescription).....	7
6.2.1 Validity Information (ValidityInformation).....	8
6.2.2 Publisher Information (PublisherInformation).....	8
6.2.3 Area Applicability (Area)	8
6.2.4 Spatial Position (SpatialPosition).....	8
6.2.5 Localization Information (LocalizationInformation).....	9
6.3 Address Signature (AddressSignature)	9
6.4 Address Class Description (AddressClassDescription).....	9
6.5 Data Types.....	9
6.5.1 Primitive Data Types.....	9
6.5.2 User Defined Data Types.....	9
6.5.3 Data Type Constraints.....	10
6.6 Address Component Description (AddressComponentDescription).....	10
6.7 Address Instance (AddressInstance).....	10
6.8 Address Component Instance (AddressComponentInstance)	10
6.9 Address Capability (AddressCapability).....	11
6.10 Address Display Template (AddressDisplayTemplate).....	11
6.11 Address Input Template (AddressInputTemplate).....	11
7.1 General.....	11
7.2 Address Specification Level	12

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 211, Geographic information / Geomatics, *Workgroup 7*.

Introduction

Addresses are among the most commonly exchanged information on the Internet, and the interchange of them is crucial to a number of Internet applications, such as electronic commerce, contact exchange, non-postal deliveries, as well as location scheduling.

Yet, addresses mean much more than just geolocation information:

- As an identity, such as an office address
- As reference points (waypoints) in routing information
- As a delivery point

The lifecycle of an address entry exchanged on the Internet typically starts with manual input of a human actor. This data, structured or unstructured, is then submitted to an Internet-connected application, and the application may in turn transmit this information to other applications or external parties on behalf of the user who provided the address. This transmission is usually performed to fulfil service delivery to the user. Within the process, there may be machine-human interactions that require display of the address in human-readable form, as well as machine-to-machine interactions on the address, such as for data validation.

Additional caution must be placed on the accuracy (or lack thereof) of human input addresses. While an address specified by a human actor may unambiguously distinguish a location, there may be intention or unintentional omissions or additions to an “official” address (if there was one).

This International Standard facilitates the interchange and interaction of addresses between humans and applications, and between applications, through the specification of a defined syntax for applications to interact and interchange international addresses, while supporting human input and human-machine interaction, without loss of fidelity.

This International Standard is part of the ISO 19160 series of standards that includes:

- ISO 19160-1:2015 — *Addressing — Conceptual model*;
- ISO 19160-2 — *Addressing — Good practices for address assignment schemes (work in progress)*;
- ISO 19160-3:2016 — *Addressing — Quality management for address data*;
- ISO 19160-4:2017 — *Addressing — International postal address components and templates*;
- ISO 19160-5 — *Addressing — Address rendering for purposes other than mail (work in progress)*.

Currently, profiles of ISO 19160-1 are published as documents (e.g. MS Word or PDF). This International Standard specifies a machine-readable encoding for such a human-readable document and for addresses conforming to such a profile.

Specifically, this part of ISO 19160 specifies a machine-readable encoding of address profiles according to ISO 19160-1 and their instances to facilitate machine usage, distribution and interoperability of addresses. In addition, the document specifies machine-readable encodings for display and input templates for addresses conforming to ISO 19160-1 address profiles outside the postal context, of which is defined in ISO 19160-4:2017. This International Standard can be the basis used for maintaining a register of profiles.

Addressing — Machine interchange syntax

1 Scope

This document specifies a machine-readable encoding for the digital storage and transmission of

- description of an address profile conforming to ISO 19160-1, *Addressing – Part 1: Conceptual model*;
- address instances that conform to a specific profile of ISO 19160-1; and
- a template for entering and displaying addresses conforming to a profile of ISO 19160-1.

2 Conformance

2.1 General

This part of ISO 19160 defines four classes of requirements and conformance. Annex A specifies how conformance with these classes shall be tested.

2.2 AddressProfileDescription

Any machine-readable description of a profile conforming to ISO 19160-1 which conformance is claimed shall pass all the requirements described in the abstract test suite in A.2.

2.3 AddressInstance

Any instance of AddressProfileDescription class for which AddressInstance conformance is claimed shall pass the requirements described in the abstract test suite in A.2.

2.4 AddressInputTemplate

Any AddressInputTemplate in address profile description model for which AddressInputTemplate conformance is claimed shall pass the requirements described in the abstract test suite in A.2.

2.5 AddressDisplayTemplate

Any AddressDisplayTemplate in address profile description model for which AddressDisplayTemplate conformance is claimed shall pass the requirements described in the abstract test suite in A.2.

3 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 19160-1, *Addressing — Part 1: Conceptual Model*

ISO 19103:2015, *Geographic information – Conceptual schema language*

ISO 639-1, *Codes for the representation of names of languages – Part 1: Alpha-2 code*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country code*

ISO 15924, *Information and documentation -- Codes for the representation of names of scripts*

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 19160-1 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <https://www.iso.org/obp>

4.1

address

structured information that allows the unambiguous determination of an object for purposes of identification and location

[SOURCE: ISO 19160-1]

4.2

addressable object

object that may be assigned an *address* (4.1)

[SOURCE: ISO 19160-1]

4.3

address class

description of a set of addresses that share the same address components, operations, methods, relationships, and semantics

[SOURCE: ISO 19160-1]

4.4

address component

constituent part of an address (4.1)

[SOURCE: ISO 19160-1]

4.5

address reference system

defined set of address components and the rules for their combination into addresses (4.1)

[SOURCE: ISO 19160-1]

4.6

lineage

provenance (4.12), source(s) and production process(es) used in producing a resource

[SOURCE: ISO 19115-1:2014, 4.9]

4.7**locale**

definition of the subset of a user's environment that depends on language and cultural conventions

[SOURCE: ISO/IEC/IEEE 9945:2009, 4.211, modified - The notes given in ISO/IEC/IEEE 9945:2009 for this entry have been omitted. The note to entry has been added.]

4.8**parent address**

address (4.1) of a *parent addressable object* (4.9)

[SOURCE: ISO 19160-1]

4.9**parent addressable object**

addressable object (4.2) that fully encloses one or more other addressable objects

[SOURCE: ISO 19160-1]

4.10**profile**

set of one or more base standards or subsets of base standards, and, where applicable, the identification of chosen clauses, classes, options and parameters of those base standards, that are necessary for accomplishing a particular function

[SOURCE: ISO 19106:2004, 4.5]

4.11**address instance**

an *address* (4.1) that is set within the context of an *address class* (4.3)

4.12**provenance**

organization or individual that created, accumulated, maintained and used records

4.13**data type**

specification of a value domain (ISO 19103:2014 4.37) with operations (ISO 19103:2014 4.26) allowed on values in this domain

[SOURCE: ISO 19103:2014, 4.14]

4.14**primitive data type**

a *data type* (4.13) that is defined in ISO 19103:2014 as "Primitive Type" (Section 7.2)

4.15**user defined data type**

a *data type* (4.13) that is defined by the user in an address profile (4.1) through the composure of other *data types* (4.13) and constraints.

4.16**user defined data type definition**

the definition of a *user defined data type* (4.15)

4.17

address capability

a marking on an *address instance* (4.11) to indicate its status and/or processes it has completed.

4.18

address component instance

TODO

4.19

address component data type

TODO

4.20

address profile validity

TODO

4.21

address display template

TODO

4.22

address input template

TODO

4.23

signature

the string of bits resulting from the signature process

[SOURCE: ISO/IEC 14888-1:2017, 4.15]

4.24

signature key

a secret data item specific to an entity and usable only by this entity in the signature process

[SOURCE: ISO/IEC 14888-1:2017, 4.18]

4.25

verification key

a data item which is mathematically related to an entity's signature key and which is used by the verifier in the verification process

[SOURCE: ISO/IEC 14888-1:2017, 4.15]

4.26

object identifier

oid

Some concrete representation for the identity of an object (instance). The object identifier (oid) is used to show examples of instances with identity, to formalize the notion of identity, and to support the notion in programming languages or database systems.

[SOURCE: ISO/IEC/IEEE 31320-2:2012, 3.1.128]

4.27

language identifier

language symbol

symbol that uniquely identifies a particular language

[SOURCE: ISO/IEC 639-3:2007, 3.3]

4.28

script

set of graphic characters used for the written form of one or more languages

[SOURCE: ISO/IEC 15924:2004, 3.7]

4.29

script code

combination of characters used to represent the name of a script

[SOURCE: ISO/IEC 15924:2004, 3.8]

4.30

URI

uniform resource identifier

[SOURCE: ISO 19103:2014, 5.3]

5 Process of Address Profile and Address Instance Interchange

TODO: Add diagrams

5.1 Address Profiles

5.1.1 Creating Address Profiles

Address profiles are created by publishers.

5.1.2 Publishing Address Profiles

Publishers distribute their address profile to others through exchange or a registry.

5.1.3 Updating Address Profiles

Publishers can update an address profile and re-distribute it by publishing using a new version number that supersedes the previously published one.

5.1.4 Using Address Profiles

Applications retrieve suitable profiles to:

- render input forms for address input according to the profile-specified InputTemplate; or
- display addresses according to the profile-specified DisplayTemplate.

Applications should consider the validity period of an address profile, and should periodically check with the publisher (or a registry that the publisher distributes via) the latest version of the address profile.

5.1.5 Retiring Address Profiles

Publisher can indicate the validity period of an address profile in the profile itself, which if the validTo date has passed, would indicate that the profile is retired.

To immediately retire an address profile, the publisher could distribute a new version of the address profile with an expired validity period.

5.2 Address Instances

5.2.1 Creating an Address Instance

This section describes how an address instance is created. The desired address profile must be already retrieved for creating an address instance that conforms to it.

Typically, a user enters an address through an application interface that implements an input format that conforms to the address profile's InputTemplate, such as an application that runs on an operating system or an Internet application. Such input interface may or may not provide a graphical form.

Conforming applications should not expect general users to be able to input an address with a fully-deduced structure. Immediately after input no capabilities are marked on the address instance.

5.2.2 Sending an Address Instance

After a user inputs an address into a structured address form, the user submits this address to the designated recipient, which could be an e-retailer, an electronic business card, or a calendar event.

5.2.3 Displaying an Address Instance

The recipient or service that receives an address instance either already has the address profile definition or should obtain the address profile definition. With the address profile definition, it could then display the address instance according to the address profile's DisplayTemplate.

5.2.4 Improving Quality of an Address Instance

The recipient or service could process the address, such as to further clean up the address by improving the conformance level by fully deducing the address structure, which would add the AddressCapability "<https://standards.iso.org/19160/-6/capabilities/specified>" to the address instance.

To ensure that the resulting address instance with a fully-specified structure is correct, the processor may wish to confirm with the user the resulting address, such as in the case where the address is used for shipping information. This would add the AddressCapability "<https://standards.iso.org/19160/-6/capabilities/confirmed>" to the address instance.

5.2.5 Verifying an Address Instance

The recipient or service could further validate the address, such as with a postal or addressing authority. This would add the AddressCapability "<https://verifyingauthority/verified>" to the address instance written by the verifying authority.

5.2.6 Adding Associated Data to an Address Instance

A processor of an address instance could add extra information such as delivery instructions or routes as associated data.

If the user already has a verified, structured address, then the service could save the effort for verifying the address. Custom AddressCapabilities could be added into the address instance.

5.2.7 Discarding an Address Instance

When a service no longer needs the address, it should be disposed of.

- Type: Uri; the intended usage of this profile
- Description: String; textual description of this definition
- TTL (time-to-live): Integer; how often should a user re-fetch this profile for updated information, in seconds.
- One or more AddressClassDescription definitions
- Zero or more UserDefinedDataType definition
- One or more AddressComponentDescription definitions
- Publisher information
- Validity information
- Localization information
- Area applicability
- Signature

6.2.1 Validity Information (ValidityInformation)

Validity is specified by:

- validTo: DateTime; the date and time when this object becomes valid.
- validFrom: DateTime; the date and time when this object becomes invalid.
- version: DateTime; issuance date/time of this object

6.2.2 Publisher Information (PublisherInformation)

Publisher information includes:

- publisherName: CharacterString; the name of the publisher;
- publisherUri: Uri; the URI that identifies the publisher, preferably an accessible location on the Internet.

6.2.3 Area Applicability (Area)

Area information includes:

- countries: ISO 3166 2-letter code[0..*]; in which jurisdictions this profile is valid
- spatial: SpatialPosition; spatial description of the applicable area

These attributes are not mutually exclusive, i.e., both can exist at the same time.

6.2.4 Spatial Position (SpatialPosition)

Spatial position information includes:

- geometry: Geometry[0..*]; ISO 19107:2017 geometry description of the location

- spatialRefSys: URI[0..1]; ISO 19107:2017 Spatial Reference System identifier (in URI form) that describes the “geometry” attribute

6.2.5 Localization Information (LocalizationInformation)

Localization information includes:

- language: ISO 639-1 2-letter code
- script: ISO 15924 4-letter Code

6.3 Address Signature (AddressSignature)

Signature information includes:

- algorithm: ISO 14888 Algorithm OID; the signature algorithm used for this signature
- publicKey: Uri; where the public key of the signer used for this signature is found
- signature: CharacterString; the signature itself encoded in Base64 format

6.4 Address Class Description (AddressClassDescription)

AddressClassDescription corresponds to the “addressClass” specified in ISO 19160-1.

An AddressClassDescription contains:

- Attributes: ID, Description
- One or more AddressComponentDescription
- One DisplayTemplate
- One InputTemplate

6.5 Data Types

6.5.1 Primitive Data Types

Primitive data types are defined in ISO 19103:2014, including:

- CharacterString
- DateTime, Date, Time
- Number, Integer, Decimal, Real
- Vector
- Boolean

6.5.2 User Defined Data Types

TODO: How to define a user defined data type?

User defined data types are defined within the profile that utilizes them. A user defined data type can be composed of multiple primitive data types, and constraints can be set on them.

- coreType: AddressComponentDataType
- constraints: DataTypeConstraints

6.5.3 Data Type Constraints

Constraints that can be set depend on the underlying data type. For example, the maximum number of Post Office Box numbers.

- Number types support “maxValue” and “minValue”.
- The “CharacterString” data type supports “maxLength” and “minLength”.
- TODO: More

As a constraint, it is also possible to provide a fixed list of accepted values to this type, such as the number of states in a country or districts in a city.

6.6 Address Component Description (AddressComponentDescription)

AddressComponentDescription corresponds to the addressComponent defined in ISO 19160-1.

- Attributes: key, description
- dataType: AddressComponentDataType; one or more

6.7 Address Instance (AddressInstance)

An instance of an address that conforms to an address profile, it contains:

- profileId: Uri
- components: AddressComponentInstance one to many
- signature: AddressSignature
- cap: AddressCapability
- spatialPosition: SpatialPosition; spatial location information of an address instance
- validity: ValidityInformaion; validity period of an address instance

6.8 Address Component Instance (AddressComponentInstance)

An instance of an address component that belongs to an AddressInstance, it contains:

- type: CharacterString; the name of the AddressComponentDescription it conforms to
- values: DataTypeInstance; the values of an AddressComponentDataType, depending on the type this may be one or multiple values.

6.9 Address Capability (AddressCapability)

A flag tagged on an AddressInstance to mark that it has fulfilled certain criteria:

- capability: Uri; what capability has this instance fulfilled
- signature: AddressSignature; to ensure that the capability is marked by someone that is trusted tfor verification of this capability

6.10 Address Display Template (AddressDisplayTemplate)

This section should be filled in.

A show template indicates how an address instance belonging to an addressClass should be shown to a user. This will be based on a subset of PADTL and may contain further extensions.

This may be performed using SVG for its layout capabilities.

The resulting language should fulfil these criteria:

- Simple to use and easy to understand
- Not verbose like XML
- Allows flexible formatting, such as hiding certain lines when there is not enough space
- Can be specified by the profile authority, as well as by the address owner (“my address should be shown this way”).

Each DisplayTemplate should also provide an example for display purposes (e.g., Help section).

DisplayTemplates should support RTL languages as well as horizontal and vertical layouts.

6.11 Address Input Template (AddressInputTemplate)

This section shows how an Input Form can be rendered according to the addressClass requirements. It should be assumed that users will not be often able enter an address thoroughly in the correct, detailed structure. Therefore, the form should be easy to understand and take into account that easy addressClass switching is important.

An application that adheres to this document may find that it is easiest for the user to first select the appropriate addressClass, then allowing users to (or automatically) add necessary components as they type.

A sample input should be provided for illustration purposes.

InputTemplates should support RTL languages as well as horizontal and vertical layouts.

This section has to be completed.

7 Address Instances

7.1 General

This section has to be completed.

7.2 Address Specification Level

When an address is first manually entered by a common person, it is unrealistic to expect this person to fully adhere to the defined structure.

There are three states of a structured address after being input:

- Unstructured. The user has entered free-form text with no regard of structuring them.
- Partially structured. Country, city and postcodes may be listed correctly, but the other components may be listed in the same row due to previous habits.
- Fully structured. All components are accurately separated and cannot be further split.

Therefore, an organization who accepts these addresses may wish to re-structure them or fill in any missing address components.

The status of the address can be specified within an address instance through the Address Capabilities structure, with the performer optionally providing a signature.

Annex A (informative)

Abstract test suites

A.1 Introduction

The abstract test suites for the conformance classes defined by this part of ISO 19160 are presented in **A.X to A.Y.**

A.2 Conformance class: AddressProfileDescription

Table A.1 — AddressProfileDescription test 1: Associations

Test purpose	Check that the model contains the associations as specified.
Test method	Inspect the model
Reference	6.2
Test type	Basic

Table A.2 — AddressProfileDescription test 2: Attributes

Test purpose	For each class and type in the model, check that the model appropriately includes the mandatory, optional and conditional attributes.
Test method	Inspect the model
Reference	6.2
Test type	Basic

A.3 Conformance class: AddressInstance

Table A.3 — AddressInstance test 1: Attributes

Test purpose	For each instance in the class, check that the instance appropriately includes the mandatory, optional and conditional attributes.
Test method	Inspect the model

Reference	6.7
Test type	Basic

A.4 Conformance class: AddressInputTemplate

Table A.4 — AddressInputTemplate test 1: Attributes

Test purpose	For each class and type in the model, check that the model appropriately includes the mandatory, optional and conditional attributes.
Test method	Inspect the model
Reference	6.10
Test type	Basic

A.5 Conformance class: AddressDisplayTemplate

Table A.5 — AddressDisplayTemplate test 1: Attributes

Test purpose	For each class and type in the model, check that the model appropriately includes the mandatory, optional and conditional attributes.
Test method	Inspect the model
Reference	6.11
Test type	Basic

Annex B (informative)

Usage

B.1 Accuracy and Verification

An authority, such as the local post office, could “verify” a structured address that it is confirmed that this address instance is a “deliverable address” through an AddressCapability.

The owner of the address, such as the tenant of an office, could provide its signed, structured address on an electronic business card. This allows the recipient of the business card to know whether the senders address is authentic. If this address is verified to be a “deliverable address”, the recipient will know that items sent to this address will very likely be deliverable.

Extra steps need to be taken here to allow this.

B.2 Address As Identity

Addresses do not only specify a location, in some cases they are part of the identity. For example, in business cards, an address can mean more than just an address, such as with vanity value.

This document must support this functionality for it to be useful in contact exchange.

// TODO: This is a comment

// As an example, the following address while complete, is more difficult to read / remember compared with the intentionally incomplete one below.

Suites 1107-1111,
Floor 11,
Central Building,
1-3 Pedder Street,
Central,
Central & Western District,
Hong Kong Island,
Hong Kong

Preferred address, as equally understandable (and deliverable):

Suite 1111,
1 Pedder Street,
Central,
Hong Kong

// TODO: This ends the comment

B.3 Address As Destination

In certain cases, an address is expected to be reachable either by person and/or post.

An address instance should support being used in conjunction with routing information, acting as a waypoint, and/or supporting a source-defined route.

For example, written instructions on how to deliver to a place that is unambiguous but terribly difficult to locate.

E.g., some buildings have split floors -- rooms may have the same floor identifier, but is actually inaccessible from the same floor.

Annex C (informative)

Examples Of Objects Specified In This Document

C.1 AddressProfileDescription

An AddressProfileDescription can be specified in the following object structure (in JSON)

```
{
  id: "http://www.iso.org/tc211/tc211-sample.adp",
  type: "iso-19160-address-profile",
  publisher: "http://www.iso.org/tc211/",
  signature: "...",

  name: "TC 211 Minimal Address Profile",
  localization: {
    language: "en",
    script: "en",
  },

  area: {
    countries: ["uk"],
  },

  dataTypes: {
    addressedObjectIdentifier: {
      primitiveType: Integer,
    }
  }
  addressComponents: { ... }
  addressClasses: { ... }
}
```

C.2 AddressClassDescription

An AddressClassDescription can be specified in the following object structure (in JSON)

```
addressClasses: {
  streetAddress: {
    description: Street Address,
    availableFields: [ (addressComponent) 1..* ... ]
    displayTemplate: { ... },
    inputTemplte: {}
  }
}
```

C.3 ValidityInformation

```
validity: {
  validFrom: 20171129Z000000,
  validTo: 20191129Z000000
}
```

C.4 PublisherInformation

```
publisher: {  
  publisherName: UK Post Office,  
  publisherUri: https://www.post.co.uk  
}
```

C.5 LocalizationInformation

```
publisher: {  
  language: en,  
  script: Latn  
}
```

C.6 AddressSignature

```
signature: {  
  algorithm: 1.2.3.4.5.6.7.8.9,  
  publicKey: https://www.post.co.uk/profile-signature.key,  
  signature: BOLVMNoGNM1TLglnlxgm0a9t  
}
```

C.7 AddressClassDescription

```
addressClassDescription: {  
  id: streetAddress,  
  description: A typical street address,  
  addressComponents: [ ... ],  
  displayTemplate: { ... },  
  inputTemplate: { ... },  
}
```

C.8 User Defined Data Types

```
dataTypes: {[  
  name: addressNumberValue,  
  coreType: Integer,  
  constraints: [ ... ],  
]}
```

C.9 Data Type Constraints

```
constraints: [{  
  maxValue: 10000,  
  minValue: 1,  
}]
```

C.10 AddressComponentDescription

```
addressComponentDescription: {  
  key: addressNumber,  
  description: Street number,  
  datatype: addressNumberValue  
}
```


C.11 AddressInstance

```
addressInstance: {
  profileId: https://standards.iso.org/19160/-6/profiles/uk.adp,
  components: [ ... ],
  signature: { ... },
  cap: [ ... ]
}
```

C.12 AddressComponentInstance

```
addressComponentInstance: {
  type: addressNumber,
  values: [ 1001 ]
}
```

C.13 AddressCapability

```
addressCapability: {
  capability: https://standards.iso.org/19160/-6/capabilities/specified,
  signature: [ ... ]
}
```

C.14 Address Display Template (AddressDisplayTemplate)

This section should be filled in.

```
addressDisplayTemplate: {
  ...
}
```

C.15 Address Input Template (AddressInputTemplate)

This section should be filled in.

```
addressInputTemplate: {
  ...
}
```

Annex D

(informative)

Examples

D.1 Example of address profiles defined in ISO 19160-1

D.1.1 ISO 19160-1 C2

```

profile = {
    id: "http://www.iso.org/tc211/tc211-minimal.adp",
    type: "iso-19160-address-profile",
    publisher: "http://www.iso.org/tc211/",
    signature: "...",

    name: "TC 211 Minimal Address Profile",
    locale: {
        language: "en",
        script: "en",
    },

    addressComponents: {
        addressLine: {
            dataType: CharacterString,
            minCardinality: 1,
            maxCardinality: n
        },
    },

    addressClasses: {
        minimalAddress: {
            availableFields: [
                {
                    componentType: addressLine,
                    min: 1,
                    max: n,
                    description: "One line of this address",
                    require: true,
                }
            ],
        },
        displayTemplates: [
            {
                /* TODO */
                orientation: horizontal,
                text: "({{ addressLine }}\n)*"
            }
        ]
    }
}
-----

```

Address Instance

```

-----
addressInstance1 = {
  profile: "http://www.iso.org/tc211/tc211-minimal.adp",
  components: [
    {
      type: addressLine,
      value: 14 Church Street,
    },
  ],
}

```

```

    {
      type: addressLine,
      value: Hatfield
    },
    {
      type: addressLine,
      value: South Africa
    }
  ]
}

addressInstance2 = {
  profile: "http://www.iso.org/tc211/tc211-minimal.adp",
  components: [
    {
      type: addressLine,
      value: Statue of Liberty
    },
    {
      type: addressLine,
      value: Liberty Island
    },
    {
      type: addressLine,
      value: New York
    },
    {
      type: addressLine,
      value: NY
    }
  ]
}

```

D.1.2 ISO 19160-1 C3

```

----
profile = {

  id: "http://www.iso.org/tc211/tc211-sample.adp",
  type: "iso-19160-address-profile",
  publisher: "http://www.iso.org/tc211/",
  signature: "...",

  name: "TC 211 Minimal Address Profile",
  locale: {
    language: "en",
    script: "en",
  },

  dataTypes: {
    addressNumberValue: {
      primitiveType: Integer,
      maxValue: 10000,
      minValue: 1,
    },
    boxNumberValue: {
      primitiveType: Integer,
      maxValue: 100000,
      minValue: 1,
    }
  }

  addressComponents: {
    addressNumber: {
      dataType: addressNumberValue,
    },
    boxNumber: {
      dataType: boxNumberValue,
    }
  }
}

```

```

    },
    /* Table C.4. Address component type */
    thoroughfareName: {
        dataType: thoroughfareNameValue,
    },
    localityName: {
        dataType: CharacterString,
    },
    postOfficeName: {
        dataType: CharacterString
    },
    postCode: {
        dataType: CharacterString
    },
    countryName: {
        dataType: thoroughfareName,
    }
    addressNumber: {
        dataType: addressedObjectIdentifier,
    },
},
addressClasses: {
    streetAddress: {
        description: Street Address,
        availableFields: [
            {
                componentType: addressNumber,
                minCardinality: 1,
                maxCardinality: 1,
                required: true,
            },
            {
                componentType: thoroughfareName,
                minCardinality: 1,
                maxCardinality: 1,
                required: true,
            },
            {
                componentType: placeName
                dataType: CharacterString,
                minCardinality: 1,
                maxCardinality: 1,
                required: true,
            },
            {
                componentType: postCode
                minCardinality: 1,
                maxCardinality: 1,
                required: true,
            },
            {
                componentType: countryName
                minCardinality: 1,
                maxCardinality: 1,
                required: false,
            },
        ],
        displayTemplates: [
            {
                /* TODO */
            }
        ]
    },
    boxAddress: {
        availableFields: [
            {
                componentType: boxNumber,
                minCardinality: 1,
            }
        ]
    }
}

```

```

        maxCardinality: 1,
        required: true,
    },
    {
        componentType: postOfficeName,
        dataType: CharacterString,
        minCardinality: 1,
        maxCardinality: 1,
        required: true,
    },
    {
        componentType: postCode
        minCardinality: 1,
        maxCardinality: 1,
        required: true,
    },
    {
        componentType: countryName
        minCardinality: 1,
        maxCardinality: 1,
        required: false,
    },
],
displayTemplates: [
    {
        /* TODO */
    }
]
}
}
```

Address Instance

```

-----
addressInstance1 = {
  profile: "http://www.iso.org/tc211/tc211-sample.adp#streetAddress",
  components: [
    {
      type: addressNumber,
      value: 99
    },
    {
      type: thoroughfareName,
      value: {
        name: Lombardy,
        type: Street
      }
    },
    {
      type: placeName,
      value: The Hills,
    },
    {
      type: postCode,
      value: 0039,
    },
    {
      type: countryName,
      value: South Africa
    }
  ]
}

boxInstance1 = {
  profile: "http://www.iso.org/tc211/tc211-sample.adp#boxAddress",
  components: [

```

```
{
  type: boxNumber,
  value: 345
},
{
  type: postOfficeName,
  value: Orlando,
},
{
  type: postCode,
  value: 2020
},
{
  type: countryName,
  value: South Africa
}
]
```

Bibliography

- [1] ISO #####-#, *General title — Part #: Title of part*
- [2] ISO #####-#:20##, *General title — Part ##: Title of part*