United Nations Economic Commission for Europe

United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT)

Core Components
Data Type Catalogue
Version 3.1
17 October 2011

Abstract

CCTS 3.0 defines the rules for developing Core Data Types and Business Data Types to define the value domains for Basic Core Components Basic Core Component Properties, Basic Business Information Entities, and Basic Business Information Entity Properties. CCTS 3.0 also stipulates that UN/CEFACT will publish a comprehensive list of approved CDTs and BDTs. This Data Type Catalogue meets that requirement. It contains the UN/CEFACT defined Core Data Types (CDTs). BDTs will be published in UN/CEFACT Directory Releases as part of the Core Component Library. This Catalogue also contains a detailed listing and explanation of the underlying primitive types used by the data types. Additionally, the XML Schema Definition (XSD) of the implied data types are provided as a hypertext link in appendix A. Appendix B defines the UN/EDIFACT mappings. This catalogue will be maintained by the UN/CEFACT Applied Technologies Group (ATG) using the data maintenance request (DMR) procedures for data types contained in Appendix C.

Contents

1 Status of This Document

This UN/CEFACT Data Type Catalogue is developed in accordance with the Trade R650 Revision 4 Open Development Process (ODP) for technical specifications. The Applied Technologies Group has approved it for public distribution as an approved specification.

This document contains information to guide in the interpretation or implementation.

The document formatting is based on the Internet Society’s Standard RFC format.

Distribution of this document is unlimited.

This version: UN/CEFACT Data Type Catalogue, Version 3.1 of 17 October 2011

Previous version: UN/CEFACT Data Type Catalogue, Version 3.0 of 29 September 2009

This document may also be available in these non-normative formats: XML, XHTML with visible change markup. See also translations.

2 Disclaimer

1. UN/CEFACT draws attention to the possibility that the practice or implementation of its outputs (which include but are not limited to Recommendations, norms, standards, guidelines and technical specifications) may involve the use of a claimed intellectual property right.

2 Each output is based on the contributions of participants in the UN/CEFACT process, who have agreed to waive enforcement of their intellectual property rights pursuant to the UN/CEFACT IPR Policy (document ECE/TRADE/C/CEFACT/2010/20/Rev.2 available at http://www.unece.org/cefact/cf_docs.html or from the UNECE secretariat). UN/CEFACT takes no position concerning the evidence, validity or applicability of any claimed intellectual property right or any other right that might be claimed by any third parties related to the implementation of its outputs. UN/CEFACT makes no representation that it has made any investigation or effort to evaluate any such rights.

3. Implementers of UN/CEFACT outputs are cautioned that any third party intellectual property rights claims related to their use of a UN/CEFACT output will be their responsibility and are urged to ensure that their use of UN/CEFACT outputs does not infringe on an intellectual property right of a third party.

4. UN/CEFACT does not accept any liability for any possible infringement of a claimed intellectual property right or any other right that might be claimed to relate to the implementation of any of its outputs.

3 Introduction

The Core Components Technical Specification (CCTS) developed by UN/CEFACT provides a methodology for semantic data modeling that achieves a common understanding of data structures and message types on a syntax independent level. It identifies the rules for defining core data types to define the value domain of conceptual model simple properties, and the rules for transforming those CDTs into business data types that define the value domain of logical model simple properties. Representation terms are provided that are used as naming conventions to represent the data types.

To support these core and business data types, a set of primitives is also defined. The primitive is the basic definition of a value domain. New in CCTS 3 and this catalogue is the concept of floating primitives, wherein a core or business data type's value domain can be defined by different primitives. Additionally, the value domains may also be expressed using a coded list of values such as a currency code list, or by an identifier scheme, such as that used to define bar codes.

3.1 Related Documents

The following standards and specifications are relevant for the definition and expression of UN/CEFACT CCTS Data Types:

- UN/CEFACT Core Components Technical Specification V3.0 or later
- UN/CEFACT XML Naming and Design Rules Technical Specification V3.0 or later
- EDIFACT Directory – Published on 6 month release cycles

3.2 Contact Information

Applied Technologies Group Chair	Mark Crawford, SAP Labs LLC (U.S.), mark.crawford@sap.com
Data Type Catalogue Project Lead	Serge Cayron, ACORD

4 CCTS Data Type Concepts

A data type defines the value domain – set of valid values – that can be used for a particular basic core component (BCC) property or basic business information entity (BBIE) property.

There are two categories of data Types (DTs)

- Core Data Type (CDT)
- Business Data Type (BDT)

Core Data Types are used with BCC properties, and Business Data Types are used with BBIE properties. Core Data Types have a content component which carries the actual data, and supplementary components which provide metadata that refine the value domain. Both content and supplementary components have one or more value domains. Each value domain is defined by a primitive or an identifier scheme or a code list. The primitive is always from an allowed set of primitives for a particular data type term. Each primitive has a set of allowed facets that serve to further refine the primitive.

Business Data Types are used with BBIE properties. Business Data Types are derived from Core Data Types, and have the same structure as their parent CDT. A BDT can be without restrictions on its parent CDT, or it can further refine the CDT through semantic restrictions in the form of Data Type Term qualifiers and context driven value domain restrictions. Value domain restrictions are expressed as either restrictions to the allowed facets of the primitive, or modifications to the scheme or list.

See Figure 3-1 for the CCTS Data Type metamodel.

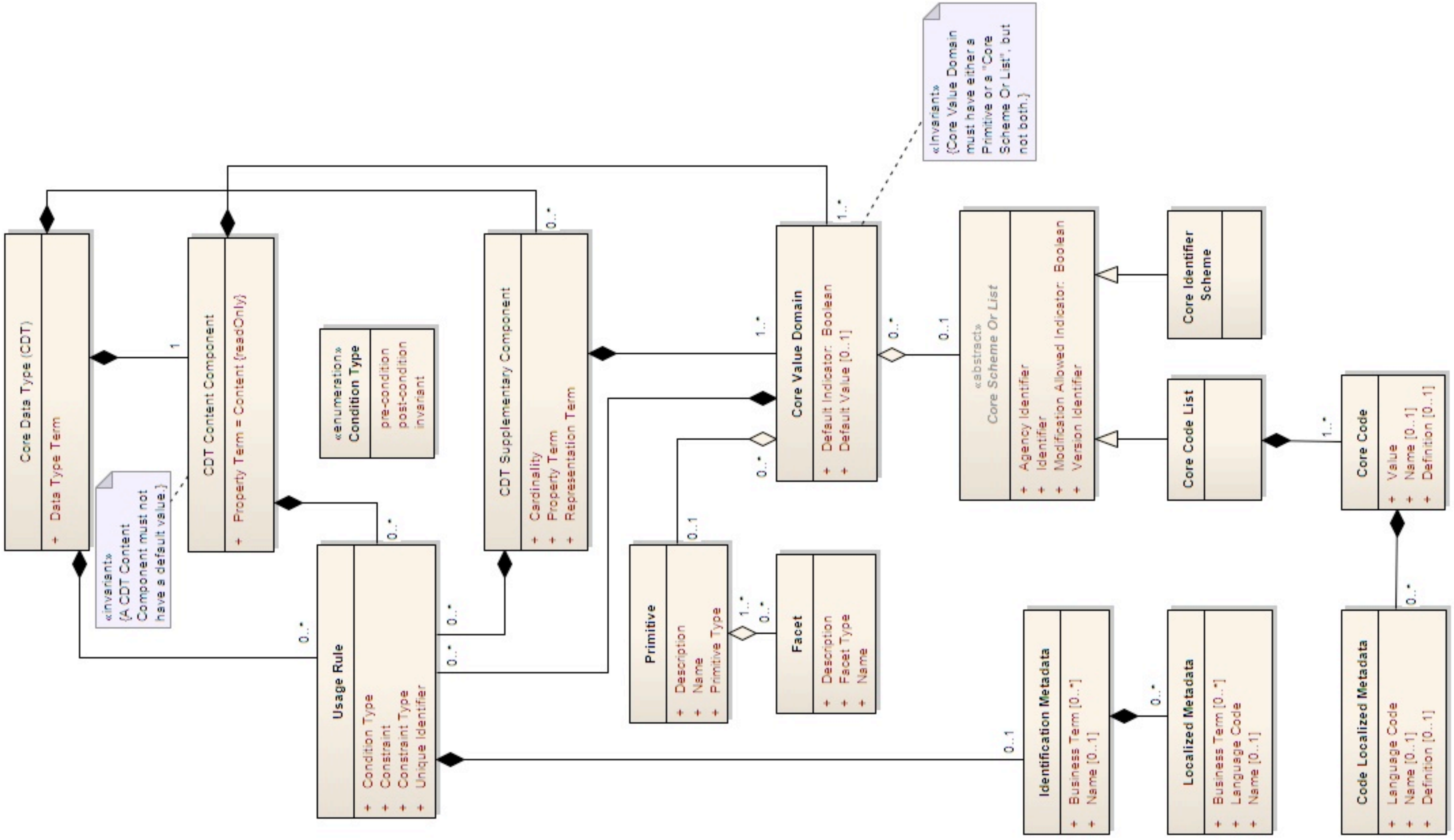

Figure 3-1 – CCTS Data Type Metamodel

4.1 Value Domains

Value domains define the set of allowed values for content and supplementary components. Value domains can be defined by either a primitive, or a code list or identifier scheme. Each content or supplementary component can have one or more value domains. When only one value domain is defined for a content or supplementary component, it will be assigned a default indicator value = **true**. When more than one value domain is defined for a content component, one will be assigned a default indicator value = **true**. When more than one value domain is defined for a supplementary component, one will be assigned a default indicator value = **true**.

4.2 Primitive Types

A primitive type, also known as a base type or built-in type, is the basic building block for the representation of a value as expressed by more complex data types. UN/CEFACT has defined a finite set of primitive types to be used by CDT and BDT content and supplementary components. Each primitive type has a set of allowed facets. Table 3-1 contains the list of allowed primitives and their facets. The facets are further defined in Table 3-2.

4.2.1 Allowed Primitives

Primitive Type	Name	Description	Allowed Facets	Remarks
Binary	Binary	Binary is a finite sequence of binary digits (bits)	Enumeration Length Minimum Length Maximum Length Pattern	
Boolean	Boolean	Boolean denotes a logical condition through a predefined enumeration of the literals true (The Boolean condition is satisfied) and false (The Boolean condition is not satisfied).	None	Allowed literals = [**true**/**false**]

Primitive Type	Name	Description	Allowed Facets	Remarks
Decimal	Decimal	Decimal is a subset of the real numbers, which can be represented by decimal numerals	Enumeration Fractional Digits Minimum Inclusive Maximum Inclusive Minimum Exclusive Maximum Exclusive Pattern Total Digits	
Double	Double	Double is the IEEE double precision 64 bits floating point type	Enumeration Minimum Inclusive Maximum Inclusive Minimum Exclusive Maximum Exclusive Pattern	
Float	Float	Float is the IEEE simple precision 32 bits floating point type	Enumeration Minimum Inclusive Maximum Inclusive Minimum Exclusive Maximum Exclusive Pattern	
Integer	Integer	Integer is a value in the infinite set (...-2, -1, 0, 1, 2...), a denumerably infinite list.	Enumeration Minimum Inclusive Maximum Inclusive Minimum Exclusive Maximum Exclusive Pattern Total Digits	
NormalizedString	Normalized String	Normalized string is a string that does not contain the carriage return (#xD), line feed (#xA) nor tab (#x9) characters.	Enumeration Length Minimum Length Maximum Length Pattern	

Primitive Type	Name	Description	Allowed Facets	Remarks
String	String	String is a sequence of characters in some suitable character set	Enumeration Length Minimum Length Maximum Length Pattern	
TimeDuration	TimeDuration	TimeDuration identifies a length of time in various time units as used in the Gregorian calendar: year, month, week, day, hour, minute, second, and fractions thereof.	Enumeration Minimum Inclusive Maximum Inclusive Minimum Exclusive Maximum Exclusive Pattern	The value domain consists of the denumerably infinite set of all possible lengths of time with the precision determined by the lowest order time unit or a fraction thereof. The value may be approximate if the Year or Month time units are used and the duration is not situated in time by a start or an end date or by context. A character string literal value that conforms to ISO 8601-2000. The TimeDuration literal denotes the TimeDuration value specified by the character string as interpreted under ISO 8601-2000.
TimePoint	TimePoint	TimePoint is a point in time to various common resolutions: year, month, week, day, hour, minute, second, and fractions thereof.	Enumeration Minimum Inclusive Maximum Inclusive Minimum Exclusive Maximum Exclusive Pattern	The value domain consists of the denumerably infinite set of all possible points in time with the resolution determined by the lowest order time unit or a fraction thereof. A character string literal value that conforms to ISO 8601-2000. The TimePoint literal denotes the TimePoint value specified by the character string as interpreted under ISO 8601-2000.

Primitive Type	Name	Description	Allowed Facets	Remarks
Token	Token	A token is a string that does not contain the line feed (#xA) nor tab (#x9) characters, that have no leading or trailing spaces (#x20) and that have no internal sequences of two or more spaces.	Enumeration Length Minimum Length Maximum Length Pattern	

Allowed Primitive Facets

Facet Type	Facet Name	Description	Value
Enumeration	Enumeration	Defines a specified set of values	A set of values from the value domain of the data type.
FractionalDigits	Fractional Digits	Defines the maximum number of fractional digits to be used.	Non Negative Integer
Length	Length	Defines the number of units of length of the data type.	Non Negative Integer
MaximumExclusive	Maximum Exclusive	Defines the upper limit of the range of allowed values. The upper limit is no allowed value. [Note] This format restriction shall not be used in combination with the Maximum Inclusive format restriction.	Value from the value domain of the data type
MaximumInclusive	Maximum Inclusive	Defines the upper limit of the range of allowed values. The upper limit is also an allowed value.	Value from the value domain of the data type
MaximumLength	Maximum Length	Defines the maximum number of units of length. [Note] This format restriction shall not be used in combination with the Length format restriction	Non Negative Integer
MinimumLength	Minimum Length	Defines the minimum number of units of length. [Note] This format restriction shall not be used in combination with the Length format restriction.	Non Negative Integer
MinimumExclusive	Minimum Exclusive	Defines the lower limit of the range of allowed values. The lower limit is no allowed value. [Note] This format restriction shall not be used in combination with the Minimum Inclusive format restriction.	Value from the value domain of the data type
MinimumInclusive	Minimum Inclusive	Defines the lower limit of the range of allowed values. The lower limit is also an allowed value.	Value from the value domain of the data type

Facet Type	Facet Name	Description	Value
Pattern	Pattern	Defines a constraint on the lexical space of a datatype to literals in a specific pattern.	Regular Expression
TotalDigits	Total Digits	Defines a maximum number of digits to be used.	Positive Integer

4.3 Lists and Schemes

Code lists and identifier schemes can be used to define the value domain of both content and supplementary components. Code lists and identifier schemes should be defined at model design time to ensure consistency and maximize interoperability.

When defined as a core value domain, a code list or identifier scheme will include the following:

- List or Scheme Name (0..1) – The name of the code list or identifier scheme
- List or Scheme Identifier (1..1) – a unique identifier for the code list or identifier scheme
- Version Identifier (1..1) – the version of the code list or identifier scheme
- Agency Identifier (1..1) – A unique identifier for the agency that owns the code list or identifier scheme
- Allowed Primitives (1..*) – The primitive that defines the value domain for the allowed code or list values
- Modification Allowed Indicator (1..1) – An indicator that defines if changes to the code list or identifier scheme are allowed
- Core Value Domain Default Indicator (1..1) – An indicator that defines if the code list or identifier scheme is the default value domain
- Core Value Domain Default Value (0..1) – A default value. The default value can be overridden.

4.4 Allowed Representation Terms

As required by CCTS, a list of allowed representation terms is included. These representation terms are used as part of BCC and BBIE Dictionary Entry Names to point to the data type that specifies their value domain.

Representation Term
Amount
Binary Object
Code
Date
Date Time
Duration
Graphic
Identifier
Indicator
Measure
Name
Number

Representation Term
Ordinal
Percent
Picture
Quantity
Rate
Ratio
Sound
Text
Time
Value
Video

See Data Type term definitions for an explanation of the representation terms.

5 Core Data Types

This section contains explicit normative expressions of core data types. These data types shall be used for all CCTS 3.0 conformant Basic Core Component properties (BCC Properties) and Basic Core Components (BCCs). They shall also be used as the basis for all CCTS 3.0 or later conformant Business Data Types.

5.1 Amount. Type

5.1.1 Data Type Term

Amount

5.1.2 Dictionary Entry Name

Amount. Type

5.1.3 Definition

An amount is a number of monetary units specified in a currency.

5.1.4 Representation Term

Amount

5.1.5 Remarks

The unit of currency may be explicit or implied.

5.1.6 Usage Guidance

Amount. Type is used to represent amounts, such as costs, remunerations, and fees.

5.1.7 Amount. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Amount. Content	Amount	Content	Decimal Double Float Integer	1..1	A number of monetary units	

5.1.8 Amount. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Amount. Currency. Code	Amount	Currency	Code	Normalized String String Token	0..1	The currency of the amount	UNDT5420SS UNDTRTB546	The primitive is specified by the Code List

5.1.9 Core Value Domains

5.1.9.1 Amount. Content

The allowed *Amount. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Decimal	True
Double	False
Float	False
Integer	False

5.1.9.2 Amount. Currency. Code

The allowed *Amount. Currency Code* core value domains consist of the following code lists:

a. ISO Codes for the representation of currencies and funds

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
ISO42173A	2009-03-05	5	Token	True	True

5.2 Binary Object. Type

5.2.1 Data Type Term

Binary Object

5.2.2 Dictionary Entry Name

Binary Object. Type

5.2.3 Definition

A binary object is a sequence of binary digits (bits).

5.2.4 Representation Term

Binary Object

5.2.5 Remarks

None

5.2.6 Usage Guidance

Binary Object. Type should be used for embedding documents such as Word, PDF and/or engineering documents. *Binary Object. Type* is differentiated from its related types - *Graphic. Type*, *Picture. Type*, *Sound. Type*, and *Video. Type*. Those types should be used where appropriate.

5.2.7 Binary Object. Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Binary Object. Content	Binary Object	Content	Binary	1..1	A finite sequence of binary digits (bits)	

5.2.8 Binary Object. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Binary Object. MIME. Code	Binary Object	MIME	Code	Normalized String String Token	0..1	The Multipurpose Internet Mail Extensions (MIME) media type of the binary object.	UNDT230W43 UNDT485R55 UNDTRTB546	Internet Engineering Task Force Request For Comments 2046 The primitive is specified by the Code List
Binary Object. Character Set. Code	Binary Object	Character Set	Code	Normalized String String Token	0..1	The character set of the binary object if the Multipurpose Internet Mail Extensions (MIME) type is text.	UNDT230W43 UNDT921934 UNDTRTB546	Internet Engineering Task Force Request For Comments 2045 The primitive is specified by the Code List
Binary Object. Filename. Name	Binary Object	Filename	Name	Normalized String String Token	0..1	The filename of the binary object	UNDTRTB546	The filename does not imply any sort of path or location dimension.

5.2.9 Core Value Domains

5.2.9.1 Binary Object. Content

The allowed *Binary Object. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Binary	True

5.2.9.2 Binary Object. MIME. Code

The allowed *Binary Object. MIME. Code* core value domains consist of the following code lists:

a. IANA MIME Media Type

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
MIMEMediaType	2009-03-04	IANA	Token	True	True

5.2.9.3 Binary Object. Character Set. Code

The allowed *Binary Object. Character Set. Code* core value domains consist of the following code lists:

a. IANA Character Sets

Code List ID [1..1]	Version ID [0..1]	Agency ID [0..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
CharacterSet	2007-05-14	IANA	Token	True	True

5.2.9.4 Binary Object. Filename. Name

The allowed *Binary Object. Filename. Name* core value domains consist of the primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	False
Token	True

5.3 Code. Type

5.3.1 Data Type Term

Code

5.3.2 Dictionary Entry Name

Code. Type

5.3.3 Definition

A code is a character string of letters, numbers, special characters (except escape sequences), and symbols. It represents a definitive value, a method, or a property description in an abbreviated or language-independent form that is part of a finite list of allowed values.

5.3.4 Representation Term

Code

5.3.5 Remarks

None

5.3.6 Usage Guidance

The *Code. Type* is used for all elements that are used in the communication between partners or systems to enable a common coded value representation. Typical example of code types are: *Country_ Code. Type* and *Language_ Code. Type*. *Code. Type* should be used in case of a finite list of allowed values and the *Identifier. Type* should be used in case of an infinite set of objects.

5.3.7 Code. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Code. Content	Code	Content	Normalized String String Token	1..1	A character string (letters, figures or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute.	UNDTRTB546

5.3.8 Code. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Code. List. Identifier	Code	List	Identifier	Normalized String String Token	0..1	The identification of a list of codes	UNDT230W43 UNDTRTB546	The primitive is specified by the Identifier Scheme
Code. List Agency. Identifier	Code	List Agency	Identifier	Normalized String String Token	0..1	The identification of the agency that manages the code list.	UNDT230W43 UNDTRTB546	The primitive is specified by the Identifier Scheme
Code. List Version. Identifier	Code	List Version	Identifier	Normalized String String Token	0..1	The identification of the version of the list of codes.	UNDT230W43 UNDTRTB546	The primitive is specified by the Identifier Scheme

5.3.9 Core Value Domain

5.3.9.1 Code. Content

The allowed *Code. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	False
Token	True

5.3.9.2 Code. List. Identifier

The allowed *Code. List. Identifier* core value domains consist of the following primitives and code lists:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	False
Token	True

5.3.9.3 Code. List Agency. Identifier

The allowed *Code. List. Agency. Identifier* core value domains consist of the following code lists:

a. UN/CEFACT Agency Identification Code

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
3055	D08B	6	Token	False	True

5.3.9.4 Code. List Version. Identifier

The allowed *Code. List Version. Identifier* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	False
Token	True

5.4 Date. Type

5.4.1 Data Type Term

Date

5.4.2 Dictionary Entry Name

Date. Type

5.4.3 Definition

A date is a Gregorian calendar representation in various common resolutions: year, month, week, day.

5.4.4 Representation Term

Date

5.4.5 Remarks

UN/CEFACT follows ISO8601 that has no format that supports UTC offset on dates. Therefore CCTS will not support this feature for dates.

5.4.6 Usage Guidance

Date. Type is used when it is only important to know the day in which something occurs, and not the time of day at which it occurs. When both the date and time are important use *Date Time. Type*. When the time zone needs to be known use *Date Time. Type*.

Date. Type should not be used to specify periodic events.

Only the Gregorian calendar will be used. The date value will be expressed in the ISO 8601-2000 defined format for date: as a combination of year, month, week and day time units. In all cases the possible values and representations of date are those defined in ISO 8601-2000 but this specification only allows a restricted set of formats to enhance interoperability.

Depending on the business context and semantics of the business data type, the following variations of date values are allowed:

- Reduced precision: a date might be reduced to the precision of the week, month or year because the date precision is either not needed or is not communicated for reason of privacy or unavailability.

- The semantics defined in ISO 8601-2000 for truncated date is refined and replaced by the semantics defined in this specification.
  - Truncation of high order units: a date might be truncated by its high order units - year, month or week - if, in the particular context of an interchange, their values are not realized.
  - Truncation of high order units is used if, in the particular context of an interchange, their values are not realized. Truncation must only be used in situations where it is possible for all communicating parties to calculate the exact dates unequivocally in other contexts where additional information is available.
    - For example an anniversary day may be communicated only with indication of the month and day in the year. An exact anniversary date can be determined (realized) for each specified year.
  - In order to promote interoperability, truncation must never be used in situation when the high order units are known in the context of the interchange.

Use of `Week` and `Ordinal` dates in particular business contexts: business may need this representation (as for truncated dates); it is up to the business to decide on their use, considering systems' interoperability. Note that since this representation format can be explicitly specified at design time or interpreted at run-time, there is no format ambiguity on a `Week` or `Ordinal` date element.

If it is important to record the time as well as the date, use the *Date Time. Type*.

5.4.7 Date. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Date. Content	Date	Content	TimePoint	1..1	The particular point in the progression of date.	UNDT2918CD

5.4.8 Date. Type Supplementary Components

None

5.4.9 Core Value Domain

5.4.9.1 Date. Content

The allowed *Date. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
TimePoint	True

5.5 DateTime. Type

5.5.1 Data Type Term

Date Time

5.5.2 Dictionary Entry Name

Date Time. Type

5.5.3 Definition

A date time identifies a date and time of day to various common resolutions: year, month, week, day, hour, minute, second, and fraction of second.

5.5.4 Representation Term

Date Time

5.5.5 Remarks

The time of day part may be expressed with or without the offset to UTC, or in UTC time.The coordinated universal time (UTC) is the standardized basis for time specifications that are used internationally. If the time of day is not expressed in UTC or has no offset to UTC then the Date Time is local.

If needed, the Time Zone. Code supplementary component can be used as an alternate way to specify the time zone to which the Date Time refers and the Daylight Saving. Indicator supplementary component can be used to specify if DateTime is in daylight saving time or not. See usage guidance below.

5.5.6 Usage Guidance

Date Time. Type is used for time stamps that should contain the day and time. For example, creation date/time, receipt date/time, processing date/time, delivery date/time, and expiry date/time.

Only the Gregorian calendar will be used.

Date and time of the day value will be expressed in the ISO 8601-2000 extended format for date and time of day: as a combination of year, month, week, day, hour, minute and second time units.

Reduced precision to `Date` is allowed, but no further reduced precision to `Month` or `Year` is allowed. If further truncation is required, use *Date. Type*. In all cases the possible values and representations of date and time are those defined in ISO 8601-2000 but this specification only allows a restricted set of formats to enhance interoperability.

Depending on the business context and semantics of the business data type, the following variations of date and time of day values are allowed:

- Reduced precision: a date and time of day might be reduced to the precision of the minute, hour or day because the time precision is either not needed or is not communicated for reason of privacy or unavailability. Fractions of the least significant time unit will not be used unless it is the second.
    - If reduced precision to Month or Year is required, use *Date. Type* only.
- The semantics defined in ISO 8601:2000 for truncated date and time is refined and replaced by the semantics defined in this specification.
    - Truncation of high order units: a date and time of day might be truncated by its high order units - year, month, week - if, in the particular context of an interchange, their values are not realized.
    - Truncation of high order units is used if, in the particular context of an interchange, their values are not realized. Truncation must only be used in situations where it is possible for all communicating parties to calculate the exact dates unequivocally in other contexts where additional information is available.
        - For example weekly air schedules may be communicated only with indication of the day in the week and time in the day in the context of a schedule planning process. The exact date and time of a trip will be determined (realized) at the time of booking, using missing information (date of trip) provided by the requestor.
    - In order to promote interoperability, truncation must never be used in situation when the high order units are known in the context of the interchange.
    - It is not allowed to truncate by the day, hour or minute units; if this is required use the *Time. Type* DT.

`Week` and **`Ordinal`** date representations may be used in particular business contexts. It is up to the business to decide on their use, considering systems' interoperability.

The Time Zone. Code supplementary component may be used as an alternate way to specify the TimeZone to which the Date Time refers. If Date Time is local (i.e. is not expressed in UTC or contains no offset to UTC) then Time Zone. Code specifies the time zone and daylight saving policy in which Date Time is expressed. If DateTime is expressed in UTC then Time Zone. Code provides the information required to calculate the corresponding local Date Time.Time Zone. Code can only be present if no offset to UTC is specified in Date Time.

The Daylight Saving. Indicator supplementary component can optionally be used in complement to Time Zone. Code to specify if DateTime is in daylight saving time or not. However the daylight saving time indicator is required in case of a local time where a duplicate hour could exist.

For example, if the local date and time (that is set in CET) is:

Oct 26th 2008 02:30

This could mean:

Oct 26th 2008 00:30 UTC or

Oct 26th 2008 01:30 UTC

In other words, if it is not clear whether the local date time is in the daylight saving time or not, the daylight saving time indicator is needed.

5.5.7 Date Time. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Date Time. Content	Date Time	Content	TimePoint	1..1	The particular date and time point in the progression of time	UNDT2918CD

5.5.8 Allowed Date Time. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Date Time. Time Zone. Code	Date Time	Time Zone	Code	Normalized String String Token	0..1	The time zone to which the date time refers	UNDT04FVC1 UNDT6N2C0S UNDT201AZX	The primitive is specified by the Code List
Date time. Daylight Saving. Indicator	Date Time	Daylight Saving	Indicator	Boolean	0..1	The indication of whether or not this Date Time is in daylight saving.		

5.5.9 Core Value Domain

5.5.9.1 Date Time. Content

The allowed *Date Time. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
TimePoint	True

5.5.9.2 Date Time. Time Zone. Code

The allowed *Date Time. Time Zone. Code* core value domains consist of the following code lists:

a. UN/CEFACT Time Zone Codes

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
TimeZoneCode	09B	6	Token	False	True

5.5.9.3 Date Time. Daylight Saving. Indicator

The allowed *Date Time. Daylight Saving. Indicator* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Boolean	True

5.6 Duration. Type

5.6.1 Data Type Term

Duration

5.6.2 Dictionary Entry Name

Duration. Type

5.6.3 Definition

A duration is the specification of a length of time without a fixed start or end time, expressed in Gregorian calendar time units (**Year**, **Month**, **Week**, **Day**) and **Hours**, **Minutes** or **Seconds**.

5.6.4 Representation Term

Duration

5.6.5 Remarks

Duration will be expressed in the ISO 8601 defined format for time intervals in years, months, weeks, days, hours, minutes, seconds, fractions of a second. Only the formats with time unit designators can be used, e.g. using the patterns defined in ISO 8601:2000

5.6.6 Usage Guidance

Duration. Type is used to represent a time interval such as scheduled, estimated, calculated or actual length of time for events or activities such as meetings, travel, vacation, or working time.

The combination of time units used in the ISO 8601 format is left to the implementer or can be restricted via patterns.

Where a length of time measured to a desired, unambiguous precision precise measurement of time is required, use *Measure. Type*.

5.6.7 Duration. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Duration. Content	Duration	Content	TimeDuration	1..1	The particular representation of duration	UNDT177117

5.6.8 Duration. Type Supplementary Components

None

5.6.9 Core Value Domains

5.6.9.1 Duration. Content

The allowed *Duration. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
TimeDuration	True

5.7 Graphic. Type

5.7.1 Data Type Term

Graphic

5.7.2 Dictionary Entry Name

Graphic. Type

5.7.3 Definition

A graphic is a diagram, a graph, mathematical curves, or similar vector based representation in binary notation (octets).

5.7.4 Representation Term

Graphic

5.7.5 Remarks

None

5.7.6 Usage Guidance

Graphic. Type is used to represent binary data and binary files for vector based graphics, such as CAD drawings, diagrams, graphs, mathematical curves and charts. Graphics may also be imbedded within other binary document formats (such as PDF, DOC, and XLS files).

Graphic. Type is differentiated from its related types – *Binary Object. Type*, *Picture. Type*, *Sound. Type*, and *Video. Type*. Those types should be used where appropriate.

5.7.7 Graphic. Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Graphic. Content	Graphic	Content	Binary	1..1	A finite sequence of binary digits (bits) for graphics.	

5.7.8 Graphic. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Graphic. MIME. Code	Graphic	MIME	Code	Normalized String String Token	0..1	The Multipurpose Internet Mail Extensions (MIME) media type of the graphic.	UNDT230W43 UNDT485R55 UNDTRTB546	Internet Engineering Task Force Request For Comments 2046 The primitive is specified by the Code List
Graphic. Character Set. Code	Graphic	Character Set	Code	Normalized String String Token	0..1	The character set of the graphic if the Multipurpose Internet Mail Extensions (MIME) type is text.	UNDT230W43 UNDT921934 UNDTRTB546	Internet Engineering Task Force Request For Comments 2045 The primitive is specified by the Code List
Graphic. Filename. Name	Graphic	Filename	Name	Normalized String String Token	0..1	The filename of the graphic	UNDTRTB546	The filename does not imply any sort of path or location dimension.

5.7.9 Core Value Domains

5.7.9.1 Graphic. Content

The allowed *Graphic. Type* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Binary	True

5.7.9.2 Graphic. MIME. Code

The allowed *Graphic. MIME. Code* core value domains consist of the following code lists:

a. IANA MIME Media Type

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
MIME Media Type	2009-03-04	IANA	Token	True	True

5.7.9.3 Graphic. Character Set. Code

The allowed *Graphic. Character Set. Code* core value domains consist of the following code lists:

a. IANA Character Sets

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
CharacterSet	2007-05-14	IANA	Token	True	True

5.7.9.4 Graphic. Filename. Name

The allowed *Graphic. Filename. Name* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	False
Token	True

5.8 Identifier. Type

5.8.1 Data Type Term

Identifier

5.8.2 Dictionary Entry Name

Identifier. Type

5.8.3 Definition

An identifier is a character string used to uniquely identify one instance of an object within an identification scheme that is managed by an agency.

5.8.4 Representation Term

Identifier

5.8.5 Remarks

There may be multiple identification schemes for identifying an object.

5.8.6 Usage Guidance

Identifier. Type is used to represent objects to enable a common identification of objects. The common identification should be based on the common identification scheme concept used to create the individual identifiers. Typical examples are "*Product_ Identifier. Type*", "*Order_ Identifier. Type*". The *"Identifier. Type"* should be used in case of an infinite set of objects, and *Code. Type* should be used in case of a finite case of allowed values.

5.8.7 Identifier. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Identifier. Content	Identifier	Content	Normalized String String Token	1..1	A character string used to uniquely identify one instance of an object within an identification scheme that is managed by an agency.	UNDTRTB546

5.8.8 Identifier. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Identifier. Scheme. Identifier	Identifier	Scheme	Identifier	Normalized String String Token	0..1	The identification of the identifier scheme.	UNDT230W43 UNDTRTB546	It is required to have common concepts for the definition of identifier scheme patterns. The primitive is specified by the Identification Scheme.
Identifier. Scheme Version. Identifier	Identifier	Scheme Version	Identifier	Normalized String String Token	0..1	The identification of the version of the identifier scheme	UNDT230W43 UNDTRTB546	The primitive is specified by the Identification Scheme.
Identifier. Scheme Agency. Identifier	Identifier	Scheme Agency	Identifier	Normalized String String Token	0..1	The identification of the agency that manages the identifier scheme	UNDT230W43 UNDTRTB546	The primitive is specified by the Identification Scheme.

5.8.9 Core Value Domains

5.8.9.1 Identifier. Content

The allowed *Identifier. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	False
Token	True

5.8.9.2 Identifier. Scheme. Identifier

The allowed *Identifier. Scheme. Identifier* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	False
Token	True

5.8.9.3 Identifier. Scheme Version. Identifier

The allowed *Identifier. Scheme Version. Identifier* core value domains consist of the following primitives:

Primitive	**Core Value Domain Default Indicator**
Normalized String	False
String	False
Token	True

5.8.9.4 Identifier. Scheme Agency. Identifier

The allowed *Identifier. Scheme Agency. Identifier* core value domains consist of the following code lists:

a. UN/CEFACT Agency Identification Codes

Code List ID [1..1]	**Version ID [1..1]**	**Agency ID [1..1]**	**Allowed Primitives**	**Modification Allowed Indicator [1..1]**	**Core Value Domain Default Indicator**
3055	D08B	6	Token	False	True

5.9 Indicator. Type

5.9.1 Data Type Term

Indicator

5.9.2 Dictionary Entry Name

Indicator. Type

5.9.3 Definition

An indicator is a list of two mutually exclusive Boolean values that express the only possible states of a property.

5.9.4 Representation Term

Indicator

5.9.5 Remarks

The allowed values are true and false.

5.9.6 Usage Guidance

Indicator. Type is used to represent binary alternatives such as classifications, indicators, and flags. The semantics of the property should reflect the Boolean concept of true or false such as *Door_ Open_ Indicator. Type* (`true|false`) and not *Door_ Position_ Indicator. Type* (which might be `open|close`). Values other than true and false should be locally mapped as appropriate.

5.9.7 Indicator. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Indicator. Content	Indicator	Content	Boolean	1..1	The value of the Indicator	UNDT39W8KS

5.9.8 Indicator. Type Supplementary Components

None

5.9.9 Core Value Domains

5.9.9.1 Indicator. Content

The allowed *Indicator. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Boolean	True

5.10 Measure. Type

5.10.1 Data Type Term

Measure

5.10.2 Dictionary Entry Name

Measure. Type

5.10.3 Definition

A measure is a numeric value determined by measuring an object along with the specified unit of measure.

5.10.4 Representation Term

Measure

5.10.5 Remarks

The unit of measure is usually required.

5.10.6 Usage Guidance

Measure. Type is used to represent a kind of physical dimension such as temperature, length, speed, width, weight, volume, latitude of an object. More precisely, *Measure. Type* should be used to measure intrinsic or physical properties of an object seen as a whole. The semantics of the physical dimension should be clearly expressed by the property term of the specific BCC. *Measure. Type* must not be confused with *Quantity. Type*.

5.10.7 Measure. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Measure. Content	Measure	Content	Decimal Double Float Integer	1..1	The numeric value determined by measuring an object.	

5.10.8 Measure. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Measure. Unit. Code	Measure	Unit	Code	**Normalized String** **String** **Token**	0..1	The unit of measure	**UNDT4862G1** **UNDTRTB546**	The primitive is specified by the Code List

5.10.9 Core Value Domains

5.10.9.1 Measure. Content

The allowed *Measure. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Decimal	True
Double	False
Float	False
Integer	False

5.10.9.2 Measure. Unit. Code

The allowed set of *Measure. Unit. Code* core value domains consist of the following code lists:

a. UN/CEFACT Codes for Units of Measure used in International Trade

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
Recommendation20	20086	6	Token	False	True

5.11 Name. Type

5.11.1 Data Type Term

Name

5.11.2 Dictionary Entry Name

Name. Type

5.11.3 Definition

A name is a word or phrase that constitutes the distinctive designation of a person, place, thing or concept.

5.11.4 Representation Term

Name

5.11.5 Remarks

A name is intended to be meaningful for human readers rather than for machines and applications.

5.11.6 Usage Guidance

Name. Type is used to represent a person, place, thing or concept.

5.11.7 Name. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Name. Content	Name	Content	Normalized String String Token	1..1	A word or phrase that represents a designation of a person, place, thing or concept.	UNDTRTB546

5.11.8 Name. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Name. Language. Code	Name	Language	Code	Normalized String String Token	0..1	The language used in the corresponding text string	UNDT29101Q UNDTRTB546	The primitive is specified by the Code List

5.11.9 Core Value Domains

5.11.9.1 Name. Content

The allowed *Name. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	False
Token	True

5.11.9.2 Name. Language. Code

The allowed *Name. Language. Code* core value domains are defined by the following schemes or lists:

a. ISO Codes for the representation of names of languages

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
ISO6392	2001-09	5	Token	False	True

5.12 Number. Type

5.12.1 Data Type Term

Number

5.12.2 Dictionary Entry Name

Number. Type

5.12.3 Definition

A mathematical number that is assigned or is determined by calculation.

5.12.4 Representation Term

Number

5.12.5 Remarks

None

5.12.6 Usage Guidance

Number. Type represents quantifying numbers as distinct from the ordinal numbers, for which *Ordinal.Type* must be used. *Number. Type* is distinguished from the other quantifying numbers – *Measure. Type and Quantity. Type* by virtue of it being dimensionless wherein the units the value represents can be inferred by its context or its parent structure. Thus *Number. Type* can only be used for truly dimension-less numbers. If the quantifying information represented is dimensionless but is either numeric or non-numeric depending on the context, *Value. Type* should be used instead.

5.12.7 Number. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Number. Content	Number	Content	Decimal Double Float Integer	1..1	Mathematical number that is assigned or is determined by calculation.	

5.12.8 Number. Type Supplementary Components

None

5.12.9 Core Value Domains

5.12.9.1 Number. Content

The allowed *Number. Content* core value domains are the following primitives:

Primitive	Core Value Domain Default Indicator
Decimal	True
Double	False
Float	False
Integer	False

5.13 Ordinal. Type

5.13.1 Data Type Term

Ordinal

5.13.2 Dictionary Entry Name

Ordinal. Type

5.13.3 Definition

An ordinal number is an assigned mathematical number that represents order or sequence.

5.13.4 Representation Term

Ordinal

5.13.5 Remarks

None

5.13.6 Usage Guidance

Ordinal. Type is used to represent ordinal numbers as distinct from the quantifying numbers. Its values are the mathematical ordinal numbers, the denumerably infinite list composed of 1, 2, 3 to infinite. The lexical value of ordinal is unsigned integer. Do not use for an indication of quantity, measure, value or amount. See also: *Number. Type*, *Value. Type*, *Ratio. Type* and *Percent. Type* for other dimensionless data types representing quantifying numbers.

5.13.7 Ordinal. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Ordinal. Content	Ordinal	Content	Integer	1..1	An assigned mathematical number that represents order or sequence	

5.13.8 Ordinal. Type Supplementary Components

None

5.13.9 Core Value Domains

5.13.9.1 Ordinal. Content

The allowed *Ordinal. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Integer	True

5.14 Percent. Type

5.14.1 Data Type Term

Percent

5.14.2 Dictionary Entry Name

Percent. Type

5.14.3 Definition

A percent is a value representing a fraction of one hundred, expressed as a quotient.

5.14.4 Representation Term

Percent

5.14.5 Remarks

Percent is dimensionless (pure number)

5.14.6 Usage Guidance

Percent. Type is used to represent a percentage that indicates how many hundredths of a basic value are to be calculated. The result of the calculation is the proportion in percent of, e.g., amounts, values, rates, discounts, and taxes.

Further examples for the application of *Percent. Type* is proportion and comparison information, such as dividends and earnings, or a percentage comparison of target and actual business results, or trade or amount margins. See also usage rules for *Rate. Type*, of which *Percent. Type* is a particular case.

5.14.7 Percent. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Percent. Content	Percent	Content	Decimal Double Float Integer	1..1	Numeric information that is assigned or is determined by percent	

5.14.8 Percent. Type Supplementary Components

None

5.14.9 Core Value Domains

5.14.9.1 Percent. Content

The allowed *Percent. Content* core value domains are the allowed set of primitives.

Primitive	Core Value Domain Default Indicator
Decimal	True
Double	False
Float	False
Integer	False

5.15 Picture. Type

5.15.1 Data Type Term

Picture

5.15.2 Dictionary Entry Name

Picture. Type

5.15.3 Definition

A picture is a visual representation of a person, object, or scene in binary notation (octets).

5.15.4 Representation Term

Picture

5.15.5 Remarks

None

5.15.6 Usage Guidance

Picture. Type should be used for embedding binary data of pictures such as photos, art, and clip art. *Picture. Type* is differentiated from its related types – *Binary Object. Type*, *Graphic. Type*, *Sound. Type*, and *Video. Type*. Those types should be used where appropriate.

5.15.7 Picture. Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Picture. Content	Picture	Content	Binary	1..1	A finite sequence of binary digits (bits) for pictures.	

5.15.8 Picture. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Picture. MIME. Code	Picture	MIME	Code	Normalized String String Token	0..1	The Multipurpose Internet Mail Extensions (MIME) media type of the picture.	UNDT230W43 UNDT485R55 UNDTRTB546	Internet Engineering Task Force Request For Comments 2046 The primitive is specified by the Code List
Picture. Character Set. Code	Picture	Character Set	Code	Normalized String String Token	0..1	The character set of the picture if the Multipurpose Internet Mail Extensions (MIME) type is text.	UNDT230W43 UNDT921934 UNDTRTB546	Internet Engineering Task Force Request For Comments 2045 The Primitive is specified by the Code List
Picture. Filename. Name	Picture	Filename	Name	Normalized String String Token	0..1	The filename of the picture	UNDTRTB546	The filename does not imply any sort of path or location dimension.

5.15.9 Core Value Domains

5.15.9.1 Picture. Content

The allowed Picture. Content core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Binary	True

5.15.9.2 Picture. MIME. Code

The allowed *Picture. MIME. Code* core value domains consist of the following code lists:

a. IANA MIME Media Types

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
MIME Media Type	2009-03-04	IANA	Token	True	True

5.15.9.3 Picture. Character Set. Code

The allowed *Picture. Character Set. Code* core value domains consist of the following code lists:

a. IANA Character Sets

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
CharacterSet	20070514	IANA	Token	True	True

5.15.9.4 Picture. Filename. Name

The allowed *Picture. Filename. Name* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	False
Token	True

5.16 Quantity. Type

5.16.1 Data Type Term

Quantity

5.16.2 Dictionary Entry Name

Quantity. Type

5.16.3 Definition

A quantity is a counted number of non-monetary units, possibly including fractions.

5.16.4 Representation Term

Quantity

5.16.5 Remarks

May include fractions.

5.16.6 Usage Guidance

Quantity. Type is used to represent a counted number of things. *Quantity. Type* should be used for simple properties of an object seen as a composite or collection or container to quantify or count its components. *Quantity. Type* should always express a counted number of things, and the property will be such as total, shipped, loaded, stored. The semantics should be clearly expressed by the property term of the BCC Property – such as `total`, `shipped`, `loaded`, `stored`. *Quantity. Type* uses a unit of measure to quantify the value of the simple property. *Quantity. Type* must not be confused with *Measure. Type*.

5.16.7 Quantity. Type Content Component

5.16.8 Quantity. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Quantity. Content	Quantity	Content	Decimal Double Float Integer	1..1	A counted number of non-monetary units possibly including fractions.	

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Quantity. Unit. Code	Quantity	Unit	Code	Normalized String String Token	0..1	The unit of measure in which the quantity is expressed	UNDT4862G1 UNDTRTB546	The primitive is specified by the Code List

5.16.9 Core Value Domains

5.16.9.1 Quantity. Content

The allowed set of core value domains are the allowed set of primitives.

Primitive	Core Value Domain Default Indicator
Decimal	True
Double	False
Float	False
Integer	False

5.16.9.2 Quantity. Unit. Code

The allowed set of *Quantity. Unit. Code* core value domains consist of the following code lists:

a. UN/CEFACT Codes for Units of Measure Used in International Trade

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
6Recommendation20	20086	6	Token	False	True

5.17 Rate. Type

5.17.1 Data Type Term

Rate

5.17.2 Dictionary Entry Name

Rate. Type

5.17.3 Definition

A rate is a quantity, amount, frequency, or dimensionless factor, measured against an independent base unit, expressed as a quotient.

5.17.4 Representation Term

Rate

5.17.5 Remarks

5.17.6 Usage Guidance

Rate. Type is used to represent the numerical quotient resulting from a comparison of two independent things such as a quantity measured with respect to another measured quantity such as a rate of speed, a measure of a part with respect to a whole - a proportion such as the mortality rate, a tax rate, the cost per unit of a commodity or service or a charge or payment calculated in relation to a particular sum or quantity such as interest rates. The quotient is calculated by taking into account the *Rate. Multiplier. Value* and *Rate. Base Multiplier. Value* supplementary components that affect the unit codes used at the numerator and the denominator. For example, if a unit of measure was expressed as hour, when what was needed was 1/1000 of an hour, the multiplier value would be `.001`. Conversely, if a unit of measure was expressed as hour, when what was needed was hundreds of hours; the multiplier value would be `100`. The input of positive and negative numbers is possible. A minus sign (`-`) must precede a negative number. A plus sign (`+`) may precede a positive number.

Rate. Type should not be confused with Percentage Type for percentages, Ratio. Type for ratios, or Measure. Type for measures – depending on how the measure was obtained and what is the intended use. However, it is acceptable to use *Rate. Type* for cases where the business requirements cannot be restricted to these sub-cases and require the flexibility of *Rate. Type*. For example *Interest. Rate* may, depending on the business context be expressed in percent such as per thousand or per million.

5.17.7 Rate. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Rate. Content	Ratio	Content	Decimal Double Float Integer	1..1	The numerical value of the rate.	UNDTWERO12

5.17.8 Rate. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Rate. Multiplier. Value	Rate	Multiplier	Value	Decimal Double Float Integer	0..1	The multiplier of the Rate. Unit. Code or Rate. Currency. Code	UNDT0123EW	Default value = 1
Rate. Unit. Code	Rate	Unit	Code	Normalized String String Token	0..1	The unit of measure of the numerator	UNDT4862G1 UNDTQAZ8XC UNDTRTB546	The primitive is specified by the Code List
Rate. Currency. Code	Rate	Currency	Code	Normalized String String Token	0..1	The currency of the numerator	UNDTQAZ8XC UNDT5420SS UNDTRTB546	The primitive is specified by the Code List
Rate. Base Multiplier. Value	Rate	Base Multiplier	Value	Decimal Double Float Integer	0..1	The multiplier of the Rate. Base Unit. Code or Rate. Base Currency. Code	UNDT0123EW	Default value = 1
Rate. Base Unit. Code	Rate	Base Unit	Code	Normalized String String Token	0..1	The unit of measure of the denominator	UNDT4862G1 UNDT7B6N33 UNDTRTB546	The primitive is specified by the Code List
Rate. Base Currency. Code	Rate	Base Currency	Code	Normalized String String Token	0..1	The currency of the denominator	UNDT7B6N33 UNDT5420SS UNDTRTB546	The primitive is specified by the Code List

5.17.9 Core Value Domains

5.17.9.1 Rate. Content

The allowed *Rate. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Decimal	True
Double	False
Float	False
Integer	False

5.17.9.2 Rate. Multiplier. Value

The allowed *Rate. Multiplier. Value* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator	Core Value Domain Default Value
Decimal	True	1
Double	False	1
Float	False	1
Integer	False	1

5.17.9.3 Rate. Unit. Code

The allowed *Rate. Unit. Code* core value domains consist of the following code lists:

a. UN/CEFACT Codes for Units of Measure Used in International Trade

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
6Recommendation20	20086	6	Token	False	True

5.17.9.4 Rate. Currency. Code

The allowed *Rate. Currency. Code* core value domains consist of the following code lists:

a. ISO Codes for the representation of currencies and funds

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
ISO42173A	2009-03-05	5	Token	True	True

5.17.9.5 Rate. Base Multiplier. Value

The allowed *Rate. Base Multiplier. Value* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator	Core Value Domain Default Value
Decimal	True	1
Double	False	1
Float	False	1
Integer	False	1

5.17.9.6 Rate. Base Unit. Code

The allowed *Rate. Base Unit. Code* core value domains consist of the following code lists:

a. UN/CEFACT Codes for Units of Measure Used in International Trade

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
6Recommendation20	20086	6	Token	False	True

5.17.9.7 Rate. Base Currency. Code

The allowed *Rate. Base Currency. Code* core value domains consist of the following code lists

a. ISO Codes for the representation of currencies and funds

Code List ID [1..1]	Version ID [0..1]	Agency ID [0..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
ISO41273A	2009-03-05	5	Token	True	True

5.18 Ratio. Type

5.18.1 Data Type Term

Ratio

5.18.2 Dictionary Entry Name

Ratio. Type

5.18.3 Definition

A ratio is a relation between two independent quantities, using the same unit of measure or currency. A ratio can be expressed as either a quotient showing the number of times one value contains or is contained within the other, or as a proportion.

5.18.4 Representation Term

Ratio

5.18.5 Remarks

Ratio. Type is dimensionless (pure number).

5.18.6 Usage Guidance

Ratio. Type is used to represent a proportion between similar things. Ratio can be expressed as a quotient – such as a decimal number value of `.75` – or as a proportion where the two terms of the fraction are made explicit – such as `3:4`.

5.18.7 Ratio. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Ratio. Content	Ratio	Content	Decimal Double Float Integer String	1..1	The quotient or proportion between two independent quantities of the same unit of measure or currency	

5.18.8 Ratio. Type Supplementary Components

None

5.18.9 Core Value Domains

5.18.9.1 Ratio. Content

The allowed *Ratio. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Decimal	True
Double	False
Float	False
Integer	False
String	False

5.19 Sound. Type

5.19.1 Data Type Term

Sound

5.19.2 Dictionary Entry Name

Sound. Type

5.19.3 Definition

A sound is any form of an audio file such as audio recordings in binary notation (octets).

5.19.4 Representation Term

Sound

5.19.5 Remarks

None

5.19.6 Usage Guidance

Sound. Type should be used for embedding binary data of voice, sound and music streams. *Sound. Type* is differentiated from its related types – *Binary Object. Type*, *Graphic. Type*, *Picture. Type*, and *Video. Type*. Those types should be used where appropriate.

5.19.7 Sound. Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Sound. Content	Sound	Content	Binary	1..1	A finite sequence of binary digits (bits) for sounds.	

5.19.8 Sound. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Sound. MIME. Code	Sound	MIME	Code	Normalized String String Token	0..1	The Multipurpose Internet Mail Extensions (MIME) media type of the sound.	UNDT230W43 UNDT485R55 UNDTRTB546	Internet Engineering Task Force Request For Comments 2046 The primitive is specified by the Code List
Sound. Character Set. Code	Sound	Character Set	Code	Normalized String String Token	0..1	The character set of the sound if the Multipurpose Internet Mail Extensions (MIME) type is text.	UNDT230W43 UNDT921934 UNDTRTB546	Internet Engineering Task Force Request For Comments 2045 The primitive is specified by the Code List
Sound. Filename. Name	Sound	Filename	Name	Normalized String „String Token	0..1	The filename of the sound	UNDTRTB546	The filename does not imply any sort of path or location dimension.

5.19.9 Core Value Domains

5.19.9.1 Sound. Content

The allowed *Sound. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Binary	True

5.19.9.2 Sound. MIME. Code

The allowed *Sound. MIME. Code* core value domains consist of the following code lists:

a. IANA MIME Media Type

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
MIME Media Type	2009-03-04	IANA	Token	True	True

5.19.9.3 Sound. Character Set. Code

The allowed *Sound. Character Set. Code* core value domains consist of the following code lists:

a. IANA Character Sets

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
Character Set	2007-05-14	6	Token	True	True

5.19.9.4 Sound. Filename. Name

The allowed *Sound. Filename. Name* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	False
Token	True

5.20 Text. Type

5.20.1 Data Type Term

Text

5.20.2 Dictionary Entry Name

Text. Type

5.20.3 Definition

Text is a character string such as a finite set of characters generally in the form of words of a language.

5.20.4 Representation Term

Text

5.20.5 Remarks

None

5.20.6 Usage Guidance

None

5.20.7 Text. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Text. Content	Text	Content	Normalized String String Token	1..1	A character string generally in the form of words of a language	

5.20.8 Text. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Text. Language. Code	Name	Language	Code	Normalized String String Token	0..1	The language used in the corresponding text string	UNDT29101Q UNDTRTB546	The primitive is specified by the Code List

5.20.9 Core Value Domains

5.20.9.1 Text. Content

The allowed Text. Type core value domains are the following primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	No
String	Yes
Token	No

5.20.9.2 Text. Language. Code

The allowed *Text. Language. Code* core value domains are defined by the following schemes or lists:

a. ISO Tags for Identification of Languages

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
ISO6392	2008-11-07	5	Token	False	True

5.21 Time. Type

5.21.1 Data Type Term

Time

5.21.2 Dictionary Entry Name

Time. Type

5.21.3 Definition

Time is a time of day to various common resolutions – hour, minute, second and fractions thereof.

5.21.4 Representation Term

Time

5.21.5 Remarks

Time means a time point that occurs in some arbitrary calendar day.

Time can be represented with or without offset to UTC, or in UTC time.

The coordinated universal time (UTC) is the standardized basis for time specifications that are used internationally.

5.21.6 Usage Guidance

Time. Type is used to represent a time on any day, such as product delivery time or the start time and end time of a period of time such as the working day or lunch hour.

The time of day value will be expressed in ISO 8601-2000 defined format for time of day: as a combination of hour, minute, second and fractions thereof time units. In all cases the possible values and representations of time of day are those defined in ISO 8601-2000 but this specification only allows a restricted set of formats to enhance interoperability.

Depending on the business context and semantics of the business data type, the following variations of time of day values are allowed:

- Reduced precision: a time of day might be reduced to the precision of the minute or hour because the time precision is either not needed or is not communicated for reason of privacy or unavailability.

  - Fractions of the least significant time unit will not be used unless it is the second.
- The semantics defined in ISO 8601-2000 for truncated time of day is refined and replaced by the semantics defined in this specification.
  - Truncation of high order units: a time of day might be truncated by its high order units - hour or minute - if, in the particular context of an interchange, their values are not realized.
  - Truncation of high order units is used if, in the particular context of an interchange, their values are not realized. Truncation must only be used in situations where it is possible for all communicating parties to calculate the exact time of day unequivocally in other contexts where additional information is available.
    - For example hourly train schedules may be communicated only with indication of the minute time unit in the context of a schedule planning process. The exact time of a trip will be determined (realized) at the time of booking, using missing information (hour of trip) provided by the requestor.
  - In order to promote interoperability, truncation must never be used in situation when the high order units are known in the context of the interchange.

If it is important to record the date as well as the time, use the *Date Time. Type*.

5.21.7 Time. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Time. Content	Time	Content	TimePoint	1..1	The particular point in the progression of time	UNDT2918CD

5.21.8 Time. Type Supplementary Components

None

5.21.9 Core Value Domains

5.21.9.1 Time. Content

The allowed *Time. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
TimePoint	True

5.22 Value. Type

5.22.1 Data Type Term

Value

5.22.2 Dictionary Entry Name

Value. Type

5.22.3 Definition

A value is the concept of worth in general that is assigned or is determined by measurement, assessment or calculation.

5.22.4 Representation Term

Value

5.22.5 Remarks

None

5.22.6 Usage Guidance

Value. Type is used to represent a quantifying information as distinct from *Ordinal. Type*, which must be used to express an order or sequence. V*alue. Type* is distinguished from *Number. Type* by virtue of it being either numeric or non-numeric depending on the context or business domain. V*alue. Type* can be associated to property terms like parameter, scale, score, level and size. Like *Number. Type*, *Value. Type* is distinguished from the other numerical types – *Measure. Type*, *Quantity. Type* by virtue of it being dimensionless wherein the units the value represents can be inferred by its context or its parent structure.

5.22.7 Value. Type Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Value. Content	Value	Content	Decimal Double Float Integer NormalizedString String Token	1..1	Information that is assigned or is determined by measurement, assessment or calculation.	

5.22.8 Value. Type Supplementary Components

None

5.22.9 Core Value Domains

5.22.9.1 Value. Content

The allowed *Value. Content* core value domains are the following primitives:

Primitive	Core Value Domain Default Indicator
Decimal	True
Double	False
Float	False
Integer	False
NormalizedString	False
String	False
Token	False

5.23 Video. Type

5.23.1 Data Type Term

Video

5.23.2 Dictionary Entry Name

Video. Type

5.23.3 Definition

A video is a recording, reproducing or broadcasting of visual images on magnetic tape or digitally in binary notation (octets).

5.23.4 Representation Term

Video

5.23.5 Remarks

None

5.23.6 Usage Guidance

Video. Type should be used for embedding binary files of video such as video sequences, movies, or films. *Video. Type* is differentiated from its related types – *Binary Object. Type*, *Graphic. Type*, *Picture. Type*, and *Sound. Type*. Those types should be used where appropriate.

5.23.7 Video. Content Component

Dictionary Entry Name	Data Type Term	Property Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier
Video. Content	Video	Content	Binary	1..1	A finite sequence of binary digits (bits) for videos.	

5.23.8 Video. Type Supplementary Components

Dictionary Entry Name	Data Type Term	Property Term	Representation Term	Allowed Primitives	Cardinality	Definition	Usage Rules Unique Identifier	Comments
Video. MIME. Code	Video	MIME	Code	Normalized String String Token	0..1	The Multipurpose Internet Mail Extensions (MIME) media type of the video.	UNDT230W43 UNDT485R55 UNDTRTB546	Internet Engineering Task Force Request For Comments 2046 The primitive is specified by the Code List
Video. Character Set. Code	Video	Character Set	Code	Normalized String String Token	0..1	The character set of the video if the Multipurpose Internet Mail Extensions (MIME) type is text.	UNDT230W43 UNDT921934 UNDTRTB546	Internet Engineering Task Force Request For Comments 2045 The primitive is specified by the Code List
Video. Filename. Name	Video	Filename	Name	Normalized String String Token	0..1	The filename of the video	UNDTRTB546	The filename does not imply any sort of path or location dimension.

5.23.9 Core Value Domains

5.23.9.1 Video. Content

The allowed *Video. Content* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Binary	True

5.23.9.2 Video. MIME. Code

The allowed *Video. MIME. Code* core value domains consist of the following code lists:

a. IANA MIME Media Types

Code List ID [1..1]	Version ID [1..1]	Agency ID [1..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
MIME Media Type	2009-03-04	IANA	Token	True	True

5.23.9.3 Video. Character Set. Code

The allowed *Video. Character Set. Code* core value domains consist of the following code lists:

a. IANA Character Sets

Code List ID [1..1]	Version ID [0..1]	Agency ID [0..1]	Allowed Primitives	Modification Allowed Indicator [1..1]	Core Value Domain Default Indicator
CharacterSet	2007-05-14	6	Token	True	True

5.23.9.4 Video. Filename. Name

The allowed *Video. Filename. Name* core value domains consist of the following primitives:

Primitive	Core Value Domain Default Indicator
Normalized String	False
String	True
Token	False

6 Usage Rules

Usage rules apply to Data Type, Data Type Content Components and Data Type Supplementary Components.

Unique Identifier	Constraint	Constraint Type Code	Constraint Language Code	Condition Type (pre, post, or invariant
UNDT230W43	Use only if no scheme or list identifiable at model design time	Unstructured		invariant
UNDT485R55	Must use IANA MIME Media Type Code	Unstructured		invariant
UNDT921934	Must use IANA Character Set Code	Unstructured		Invariant
UNDT39W8KS	Must use literals [true/false]	Unstructured		Invariant
UNDT4862G1	Must use UN/CEFACT Recommendation 20	Unstructured		Invariant
UNDT29101Q	Must use Internet Engineering Task Force Request For Comments 3066 and ISO 639-2.	Unstructured		Invariant
UNDTQAZ8XC	Only one of Unit Code or Currency. Code can be specified.	Unstructured		Invariant
UNDT7B6N33	Only one of Base Unit. Code or Base Currency. Code can be specified	Unstructured		Invariant
UNDT0123EW	Use only if value is different than default	Unstructured		Invariant
UNDT5420SS	Defaults to latest version ISO 4217 3 alpha	Unstructured		Invariant
UNDTRTB546	Defaults to Token primitive	Unstructured		Invariant
UNDT2918CD	Must use an ISO 8601:2000 conformant representation as per the relevant subset defined in Appendix D	Unstructured		Invariant
UNDT177117	Must use an ISO 8601:2000 conformant Duration representation using time unit designators only	Unstructured		Invariant
UNDT09241X	\d+:\d+	Structured	RegularExpression	Invariant
UNDTWERO12	Must precede a negative number with a minus sign ‘-‘	Unstructured		Invariant
UNDT04FVC1	If Date Time contains neither offset to UTC nor Z, Date Time is local and Time Zone Code specifies the Time Zone to which Date Time refers.	Unstructured		Invariant
UNDT6N2C0S	If Date Time contains Z, Date Time is in the UTC and Time Zone Code specifies the Time Zone in which Date Time should be displayed to the user.	Unstructured		Invariant
UNDT201AZX	Time Zone. Code can only be present if no offset to UTC (+/-hh:mm) is specified in DateTime.	Unstructured		Invariant

Appendix A XSD Representation

The XSD representation of Reference BDTs may be found here:

http://www1.unece.org/cefact/platform/display/ATG/Data+Type+Catalogue+Project+Version+3.0

Appendix B UN/EDIFACT Representation

UN/EDIFACT supports a limited number of core data types as shown in the following sections.

B. 1 Amount. Type

The CDT `Amount. Type` maps to UN/EDIFACT segment `MOA Monetary Amount`. The `Amount. Type` content component maps to data element `5004 Monetary amount`. The `Amount. Currency. Code` supplementary component maps to data element `6345 Currency identification code`.

```
MOA  MONETARY AMOUNT

       Function: To specify a monetary amount.

010    C516 MONETARY AMOUNT                         M    1

       5025  Monetary amount type code qualifier    M      an..3

       5004  Monetary amount                        C      n..35

       6345  Currency identification code           C      an..3

       6343  Currency type code qualifier           C      an..3

       4405  Status description code                C      an..3
```

B. 2 Date. Type

The CDT `Date. Type` maps to UN/EDIFACT segment `DTM Date/Time/Period`. The `Date. Type` content component maps to the combination of data `element 2380 Date or time or period text` and data element `2379 Date or time or period format code`. See Section B.7 for specific expressions of data element 2379 codes as ISO 8601:2000 conformant expressions.

```
        DTM  DATE/TIME/PERIOD

       Function: To specify date, and/or time, or period.

010    C507 DATE/TIME/PERIOD                        M    1

       2005  Date or time or period function code

             qualifier                              M      an..3

       2380  Date or time or period text            C      an..35
```

```
         2379  Date or time or period format code        C       an..3
```

B. 3 Date Time. Type

The CDT **`Date Time. Type`** maps to UN/EDIFACT segment **`DTM Date/Time/Period`**. The **`Date Time. Type`** content component maps to the combination of data **`element 2380 Date or time or period text`** and data element **`2379 Date or time or period format code`**. See Section B.7 for specific expressions of data element 2379 codes as ISO 8601:2000 conformant expressions.

```
          DTM  DATE/TIME/PERIOD

         Function: To specify date, and/or time, or period.

010      C507 DATE/TIME/PERIOD                           M    1

         2005  Date or time or period function code

               qualifier                                 M       an..3

         2380  Date or time or period text               C       an..35

         2379  Date or time or period format code        C       an..3
```

B. 4 Measure. Type

The CDT **Measure. Type** maps to UN/EDIFACT segment **MEA Measurements**. The **Measure. Type** content component maps to data element **6314 Measure**. The Measure. Unit. Code supplementary component maps to date element 6411 Measure Unit Code with a fixed value source of *UN/ECE Recommendation 20 Codes for Units of Measure*.

```
MEA   MEASUREMENTS
        Function: To specify physical measurements, including
                  dimension tolerances, weights and counts.
010     6311 MEASUREMENT PURPOSE CODE QUALIFIER         M    1 an..3
020     C502 MEASUREMENT DETAILS                        C    1
        6313  Measured attribute code                   C      an..3
        6321  Measurement significance code             C      an..3
        6155  Non-discrete measurement name code        C      an..17
        6154  Non-discrete measurement name             C      an..70
030     C174 VALUE/RANGE                                C    1
        6411  Measurement unit code                     M      an..8
        6314  Measure                                   C      an..18
        6162  Range minimum quantity                    C      n..18
        6152  Range maximum quantity                    C      n..18
        6432  Significant digits quantity               C      n..2
040     7383 SURFACE OR LAYER CODE                      C    1 an..3
```

B. 5 Quantity. Type

The CDT `Quantity. Type` maps to UN/EDIFACT segment `QTY Quantity`. The `Quantity. Type` content component maps to data element `6060 Quantity`. The `Quantity. Unit. Code` supplementary component maps to date element `6411 Measure Unit Code` with a fixed value source of *UN/ECE Recommendation 20 Codes for Units of Measure*.

```
QTY  QUANTITY

       Function: To specify a pertinent quantity.

010    C186 QUANTITY DETAILS                              M    1

       6063  Quantity type code qualifier                 M      an..3

       6060  Quantity                                     M      an..35

       6411  Measurement unit code                        C      an..8
```

B. 6 Time. Type

The CDT `Time. Type` maps to UN/EDIFACT segment `DTM Date/Time/Period`. The `Time. Type` content component maps to the combination of data `element 2380 Date or time or period text` and data element `2379 Date or time or period format code`. See B.7 for specific expressions of data element 2379 codes as ISO 8601:2000 conformant expressions.

```
        DTM  DATE/TIME/PERIOD

       Function: To specify date, and/or time, or period.

010    C507 DATE/TIME/PERIOD                              M    1

       2005  Date or time or period function code

             qualifier                                    M      an..3

       2380  Date or time or period text                  C      an..35

       2379  Date or time or period format code           C      an..3
```

B. 7 Mapping Between UN/EDIFACT Data Element 2379 and ISO 8601:2000

In order to allow for translation of pre-existing UN/EDIFACT message content to Core Component representation and then to alternate syntaxes (e.g. XML), we need to assure that all the Date and Time formatted elements used in UN/EDIFACT are able to be converted into equivalent standard BBIE's and BDT's as defined in the Data Type Catalogue.

For the representation of dates, UN/EDIFACT uses the DTM Date/Time/Period Segment and Data Element (DE) 2379 date format code list. The Data Type catalogue uses a controlled subset of ISO 8601:2000 (See Appendix D). The requirement is to define mapping or transformation rules for each DE2379 code toward the Date/Time representations defined in the Data Type Catalogue.

There is no requirement for a full reverse mapping solution, i.e. to support all Data Type Catalogue defined Date/Time representations in the UN/EDIFACT syntax.

B. 7. 1 Principles of Solution

a. ISO8601:2000 is kept as the only date-time representation standard in the DT Catalogue to facilitate technology implementation

b. Mapping from UN/EDIFACT DE2379 to ISO8601 representations are specified in the form of conversion tables, which take into account the following principles:

   1. The UN/EDIFACT DE2379 representations that are semantically equivalent to one another (e.g. that only differ by the order of the date/time component) are mapped to a single ISO 8601 representation.

   2. The UN/EDIFACT DE2379 representations that use date with century truncation are mapped to full ISO 8601 date representations, assuming that the truncated dates can be transformed into full dates (to avoid the year 2000 issue).

   3. The UN/EDIFACT DE2379 representations of periods (start date/time - end date/time) are mapped to the ISO 8601 representations of their date/time constituents, assuming that the period element is split into two date/time elements.

   4. The UN/EDIFACT DE2379 representations that cannot be completely mapped to ISO8601 without loosing semantics are decomposed into constituent standard BDT's.
   The UN/EDIFACT DE2379 representations of quantities of time units are mapped to Quantity. Type BDT's with the appropriate measurement unit of time.

B. 7. 2 UN/EDIFACT DE2379 Conversion Tables

B. 7. 2. 1 UN/EDIFACT DE2379 Date Simple Representations

The table below lists the UN/EDIFACT DE2379 representations of simple dates that can be mapped to an ISO 8601 representation without loss of semantics.

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment
002	DDMMYY	*Calendar date: D = Day; M = Month; Y = Year.*		YYYY-MM-DD	Map to semantically equivalent ISO representation without century truncation
003	MMDDYY	*Calendar date: M = Month; D = Day; Y = Year.*		YYYY-MM-DD	Map to semantically equivalent ISO representation without century truncation
004	DDMMCCYY	*Calendar date C=Century; Y=Year; M=Month; D=Day.*		YYYY-MM-DD	Map to semantically equivalent ISO representation
101	YYMMDD	YYMMDD	Yes	YYYY-MM-DD	Map to semantically equivalent ISO representation without century truncation
102	CCYYMMDD	*Calendar date: C = Century ; Y = Year ; M = Month ; D = Day.*	Yes	YYYY-MM-DD	
103	YYWWD	*Calendar week day: Y = Year ; W = Week ; D = Day Week number 01 is always first week of January Day number 1 is always Monday.*		YYYY-Www-D	Map to semantically equivalent ISO representation without century truncation
105	YYDDD	*Calendar day: Y = Year ; D = Day January the first = Day 001 Always start numbering the days of the year from January 1st through December 31st.*	Yes	YY- DDD	Map to semantically equivalent ISO representation without century truncation
106	MMDD	*Day of a month: M = Month; D = Day.*	Yes	--MM-DD	
107	DDD	*Day's number within a specific year: D = Day.*	Yes	-DDD	
108	WW	*Week's number within a specific year: W = Week.*		-Www	
109	MM	*Month's number within a specific year: M = Month.*	Yes	--MM	
110	DD	*Day's number within is a specific month: D = Day.*	Yes	---DD	
600	CC	*Century.*	Yes	YY	

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment
601	YY	*Calendar year: Y = Year.*		YYYY	Map to semantically equivalent ISO representation without century truncation
602	CCYY	*Calendar year including century: C = Century; Y = Year.*	Yes	YYYY	
609	YYMM	*Month within a calendar year: Y = Year; M = Month.*	Yes	YYYY-MM	Map to semantically equivalent ISO representation without century truncation
610	CCYYMM	*Month within a calendar year: CC = Century; Y = Year; M = Month.*	Yes	YYYY-MM	
615	YYWW	*Week within a calendar year: Y = Year; W = Week 1st week of January = week 01.*		YYYY-Www	Map to semantically equivalent ISO representation without century truncation
616	CCYYWW	*Week within a calendar year: CC = Century; Y = Year; W = Week (1st week of January = week 01).*		YYYY-Www	
813	Day of the week	*Numeric representation of the day (Monday = 1).*		-W-D	

B. 7. 2. 2 UN/EDIFACT DE2379 Date Period Representations

The table below lists the UN/EDIFACT DE2379 representations of date periods (composed of a start and end date) that can be mapped to an ISO 8601 representation without loss of semantics, provided that the period element is split into two date elements.

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment
701	YY-YY	*A period of time specified by giving the start year followed by the end year (both without century). Data is to be transmitted as consecutive characters without hyphen.*		YYYY	Map to semantically equivalent ISO representation without century truncation

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment
702	CCYY-CCYY	*A period of time specified by giving the start year followed by the end year (both including century). Data is to be transmitted as consecutive characters without hyphen.*	Yes	YYYY	
709	YYMM-YYMM	*A period of time specified by giving the start month of a year followed by the end month of a year (both not including century). Data is to be transmitted as consecutive characters without hyphen.*		YYYY-MM	Map to semantically equivalent ISO representation without century truncation
710	CCYYMM-CCYYMM	*A period of time specified by giving the start month of a year followed by the end month of a year (both including century). Data is to be transmitted as consecutive characters without hyphen.*	Yes	YYYY-MM	
711	CCYYMMDD-CCYYMMDD	*Format of period to be given in actual message without hyphen.*	Yes	YYYY-MM-DD	
715	YYWW-YYWW	*A period of time specified by giving the start week of a year followed by the end week of year (both not including century). Data is to be transmitted as consecutive characters without hyphen.*		YYYY-Www	Map to semantically equivalent ISO representation without century truncation
716	CCYYWW-CCYYWW	*A period of time specified by giving the start week of a year followed by the end week of year (both including century). Data is to be transmitted as consecutive characters without hyphen.*		YYYY-Www	
717	YYMMDD-YYMMDD	*A period of time specified by giving the start date followed by the end date (both not including century). Data is to be transmitted as consecutive characters without hyphen.*		YYYY-MM-DD	Map to semantically equivalent ISO representation without century truncation
718	CCYYMMDD-CCYYMMDD	*A period of time specified by giving the start date followed by the end date (both including century). Data is to be transmitted as consecutive characters without hyphen.*	Yes	YYYY-MM-DD	

B. 7. 2. 3 UN/EDIFACT DE2379 Time Simple Representations

The table below lists the UN/EDIFACT DE2379 representations of simple times that can be mapped to an ISO 8601 representation without loss of semantics.

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment
401	HHMM	*Time without seconds: H = Hour; m = Minute.*	Yes	hh:mm	
402	HHMMSS	*Time with seconds: H = Hour; m = Minute; s = Seconds.*	Yes	hh:mm:ss	
404	HHMMSSZZZ	*See 402 plus Z=Time zone.*		hh:mm:ss+hh:mm	Map to semantically equivalent ISO representation

B. 7. 2. 4 UN/EDIFACT DE2379 Time Period Representations

The table below lists the UN/EDIFACT DE2379 representations of time periods (composed of a start and end time) that can be mapped to an ISO 8601 representation without loss of semantics, provided that the period element is split into two time elements.

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment
501	HHMMHHMM	*Time span without seconds: H = Hour; m = Minute;.*	Yes	hh:mm	
502	HHMMSS-HHMMSS	*A period of time specified by giving the start time followed by the end time (both expressed by hours minutes and seconds). Data is to be transmitted as consecutive characters without hyphen.*	Yes	hh:mm:ss	

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment
503	HHMMSSZZZ-HHMMSSZZZ	*A period of time specified by giving the start time followed by the end time (both expressed by hours minutes, seconds and time zone). Data is to be transmitted as consecutive characters without hyphen.*		hh:mm:ss+hh:mm	Map to semantically equivalent ISO representation

B. 7. 2. 5 UN/EDIFACT DE2379 Date and Time Simple Representations

The table below lists the UN/EDIFACT DE2379 representations of simple dates and times that can be mapped to an ISO 8601 representation without loss of semantics.

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment
005	DDMMCCYYHHMM	*Calendar date and time: C=Century; Y=Year; M=Month; D=Day; H=Hour; M=Minute.*		YYYY-MM-DDThh:mm	Map to semantically equivalent ISO representation
010	CCYYMMDDTHHMM	*Calendar date including time with minutes: C=Century; Y=Year; M=Month; D=Day; T=Time designator; H=Hour; M=Minutes. The character [T] shall be used as time designator to indicate the start of the representation of the time. For example: 20010912T1433.*	Yes	YYYY-MM-DDThh:mm	
201	YYMMDDHHMM	*Calendar date including time without seconds: Y = Year; M = Month; D = Day; H = Hour; M = Minute*		YYYY-MM-DDThh:mm	Map to semantically equivalent ISO representation without century truncation
202	YYMMDDHHMMSS	*Calendar date including time with seconds: Y = Year; M = Month; D = Day; H = Hour; m = Minutes = Seconds.*		YYYY-MM-DDThh:mm:ss	Map to semantically equivalent ISO representation without century truncation
203	CCYYMMDDHHMM	*Calendar date including time with minutes: C=Century; Y=Year; M=Month; D=Day; H=Hour; M=Minutes.*		YYYY-MM-DDThh:mm	

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment
204	CCYYMMDDHHMMSS	*Calendar date including time with seconds: C=Century;Y=Year; M=Month;D=Day;H=Hour; M=Minute;S=Second.*		YYYY-MM-DDThh:mm:ss	
205	CCYYMMDDHHMMZHHMM	*Calendar date including time and time zone expressed in hours and minutes. ZHHMM = time zone given as offset from Coordinated Universal Time (UTC).*		YYYY-MM-DDThh:mm+hh:mm	
301	YYMMDDHHMMZZZ	*See 201 + Z = Time zone.*		YYYY-MM-DDThh:mm+hh:mm	Map to semantically equivalent ISO representation without century truncation
302	YYMMDDHHMMSSZZZ	*See 202 + Z = Time zone.*		YYYY-MM-DDThh:mm:ss+hh:mm	Map to semantically equivalent ISO representation
303	CCYYMMDDHHMMZZZ	*See 203 plus Z=Time zone.*		YYYY-MM-DDThh:mm+hh:mm	Map to semantically equivalent ISO representation
304	CCYYMMDDHHMMSSZZZ	*See 204 plus Z=Time zone.*		YYYY-MM-DDThh:mm:ss+hh:mm	Map to semantically equivalent ISO representation
305	MMDDHHMM	*Month, day, hours, minutes; M = Month; D = Day; H = Hour; M = Minute.*		--MM-DDThh:mm	
306	DDHHMM	*Day, hours, minutes; D = Day; H = Hour; M = Minute.*		---DDThh:mm	

B. 7. 2. 6 UN/EDIFACT DE2379 Date and Time Period Representations

The table below lists the UN/EDIFACT DE2379 representations of date and time periods (composed of a start and end date and time) that can be mapped to an ISO 8601 representation without loss of semantics, provided that the period element is split into two date and time elements.

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment

Code	Format Template	Description	Matches ISO 8601 basic format	Mapping to ISO 8601 extended format	Comment
713	YYMMDDHHMM-YYMMDDHHMM	*A period of time specified by giving the start time followed by the end time (format year, month, day, hour and minute). Data is to be transmitted as consecutive characters without hyphen.*		-YY-MM-DDThh:mm	Map to semantically equivalent ISO representation without century truncation
719	CCYYMMDDHHMM-CCYYMMDDHHMM	*A period of time which includes the century, year, month, day, hour and minute. Format of period to be given in actual message without hyphen.*	Yes	YYYY-MM-DDThh:mm	
720	DHHMM-DHHMM	*Format of period to be given without hyphen (D=day of the week, 1=Monday; 2=Tuesday; ...7=Sunday).*		-W-DThh:mm	

B. 7. 2. 7 UN/EDIFACT DE2379 Date Special Representations

The table below lists the UN/EDIFACT DE2379 representations of simple dates that cannot be mapped to an ISO 8601 format without loss of semantics. The conversion of such date representations involves combining an ISO 8601 date BDT and another BDT type with own semantics and pattern facet.

Code	Format Template	Description	Partial mapping to ISO 8601 extended format	Comment
006	CCYYMMB	*Half-month: CC=century YY=year MM=month, B=1:first half month, B=2:second half month.*	YYYY-MM	Map the CCYYMM component to ISO 8601 as indicated Map the B component to an Ordinal. Type BDT with own semantics and pattern facet
007	CCYYMMW	*Week within a calendar month: CC=century YY=year MM=month. W=1-5 first week to fifth week in a month.*	YYYY-MM	Map the CCYYMM component to ISO 8601 as indicated Map the W component to an Ordinal. Type BDT with own semantics and pattern facet

Code	Format Template	Description	Partial mapping to ISO 8601 extended format	Comment
008	CCYYMMDDS	*Shift within a calendar day: CC=century YY=year MM=month DD=day S=1-9 shift in a day.*	YYYY-MM-DD	Map the CCYYMMDD component to ISO 8601 as indicated Map the S component to an Ordinal. Type BDT with own semantics and pattern facet
009	CCYYMMDDPP	*Time period within a calendar day: CC=century YY=year MM=month DD=day PP=00-99 time period*	YYYY-MM-DD	Map the CCYYMMDD component to ISO 8601 as indicated Map the PP component to an Identifier. Type BDT with own semantics and pattern facet
603	YYS	*Semester in a calendar year: Y = Year; S = Semester.*	YYYY	Map the YY component to ISO 8601 (without century truncation) as indicated Map the S component to an Ordinal. Type BDT with own semantics and pattern facet
604	CCYYS	*Semester in a calendar year: C = Century; Y = Year; S = Semester.*	YYYY	Map the CCYY component to ISO 8601 as indicated Map the S component to an Ordinal. Type BDT with own semantics and pattern facet
608	CCYYQ	*Quarter in a calendar year: C = Century; Y = Year; Q = Quarter.*	YYYY	Map the CCYY component to ISO 8601 as indicated Map the Q component to an Ordinal. Type BDT with own semantics and pattern facet
613	YYMMA	*To specifiy a ten-day period within a month of a year (A = ten day period).*	YYYY-MM	Map the YY component to ISO 8601 (without century truncation) as indicated Map the A component to an Ordinal. Type BDT with own semantics and pattern facet
614	CCYYMMA	*To specifiy a ten-day period within a month of a year, including century (A = ten day period).*	YYYY-MM	Map the CCYY component to ISO 8601 as indicated Map the A component to an Ordinal. Type BDT with own semantics and pattern facet

B. 7. 2. 8 UN/EDIFACT DE2379 date special period representations

The table below lists the UN/EDIFACT DE2379 representations of date periods (composed of a start and end date) that cannot be mapped to an ISO 8601 format without loss of semantics. The conversion is as follows:

- split the period element into two date elements,

- convert the constituent date representations into a combination of an ISO 8601 date BDT and of another BDT type with own semantics and pattern facet.

Code	Format Template	Description	Partial mapping to ISO 8601 extended format	Comment
104	MMWW-MMWW	*A period of time specified by giving the start week of a month followed by the end week of a month. Data is to be transmitted as consecutive characters without hyphen.*	-MM	Map the MM component to ISO 8601 as indicated Map the W component to an Ordinal. Type BDT with own semantics and pattern facet
703	YYS-YYS	*A period of time specified by giving the start semester of a year followed by the end semester of a year (both not including century). Data is to be transmitted as consecutive characters without hyphen.*	YYYY	Map the YY component to ISO 8601 (without century truncation) as indicated Map the S component to an Ordinal. Type BDT with own semantics and pattern facet
704	CCYYS-CCYYS	*A period of time specified by giving the start semester of a year followed by the end semester of a year (both including century). Data is to be transmitted as consecutive characters without hyphen.*	YYYY	Map the YYYY component to ISO 8601 as indicated Map the S component to an Ordinal. Type BDT with own semantics and pattern facet
705	YYPYYP	*Format of period to be given without hyphen (P = period of 4 months).*	YYYY	Map the YY component to ISO 8601 (without century truncation) as indicated Map the P component to an Ordinal. Type BDT with own semantics and pattern facet
706	CCYYP-CCYYP	*Format of period to be given without hyphen (P = period of 4 months).*	YYYY	Map the TTYY component to ISO 8601 as indicated Map the P component to an Ordinal. Type BDT with own semantics and pattern facet
707	YYQ-YYQ	*A period of time specified by giving the start quarter of a year followed by the end quarter of year (both not including century). Data is to be transmitted as consecutive characters without hyphen.*	YYYY	Map the YY component to ISO 8601 (without century truncation) as indicated Map the Q component to an Ordinal. Type BDT with own semantics and pattern facet

Code	Format Template	Description	Partial mapping to ISO 8601 extended format	Comment
708	CCYYQ-CCYYQ	*A period of time specified by giving the start quarter of a year followed by the end quarter of year (both including century). Data is to be transmitted as consecutive characters without hyphen.*	YYYY	Map the CCYY component to ISO 8601 as indicated Map the Q component to an Ordinal. Type BDT with own semantics and pattern facet

B. 7. 2. 9 UN/EDIFACT DE2379 time special representations

The table below lists the UN/EDIFACT DE2379 representations of simple times and cannot be mapped to an ISO 8601 representation without loss of semantics. The conversion of such date representations involves combining an ISO 8601 time BDT and another BDT type with own semantics and pattern facet.

Code	Format template	Description	Partial mapping to ISO 8601 extended format	Comment
405	MMMMSS	*Time without hours: m=minutes, s=seconds.*	--ss	Map the MMMM component to a Text. Type BDT with own semantics and pattern facet Map the SS component to ISO 8601 as indicated
406	ZHHMM	*Offset from Coordinated Universal Time (UTC) where Z is plus (+) or minus (-).*		This is not a time per say, but the property of a time. Map to a Text.Type BDT with own semantics and pattern facet, using the normalized ISO representation of time offset ("+hh:mm")

B. 7. 2. 10 UN/EDIFACT DE2379 quantity related representations

The table below lists the UN/EDIFACT DE2379 representations of quantities expressed in units of time, which should be mapped to Quantity. Type BDT's with the appropriate measurement unit of time.

Code	Format template	Description	Mapping to REC20 unit of measure	Comment

Code	Format template	Description	Mapping to REC20 unit of measure	Comment
801	Year	*To indicate a quantity of years.*	ANN	Map to Quantity. Type BDT and use REC20 unit of measure as indicated
802	Month	*To indicate a quantity of months.*	MON	Map to Quantity. Type BDT and use REC20 unit of measure as indicated
803	Week	*To indicate a quantity of weeks.*	WEE	Map to Quantity. Type BDT and use REC20 unit of measure as indicated
804	Day	*To indicate a quantity of days.*	DAY	Map to Quantity. Type BDT and use REC20 unit of measure as indicated
805	Hour	*To indicate a quantity of hours.*	HUR	Map to Quantity. Type BDT and use REC20 unit of measure as indicated
806	Minute	*To indicate a quantity of minutes.*	MIN	Map to Quantity. Type BDT and use REC20 unit of measure as indicated
807	Second	*To indicate a quantity of seconds.*	SEC	Map to Quantity. Type BDT and use REC20 unit of measure as indicated
808	Semester	*To indicate a quantity of semesters (six months).*	SAN	Map to Quantity. Type BDT and use REC20 unit of measure as indicated
809	Four months period	*To indicate a quantity of four months periods.*		Map to Quantity. Type BDT with own semantics and implicit unit of measure
810	Trimester	*To indicate a quantity of trimesters (three months).*	QAN	Map to Quantity. Type BDT and use REC20 unit of measure as indicated
811	Half month	*To indicate a quantity of half months.*		Map to Quantity. Type BDT with own semantics and implicit unit of measure
812	Ten days	*To indicate a quantity of ten days periods.*	DAD	Map to Quantity. Type BDT and use REC20 unit of measure as indicated
814	Working days	*Number of working days.*	E49	Map to Quantity. Type BDT and use REC20 unit of measure as indicated

Appendix C Data Maintenance Request Procedures

C. 1 Criteria For Creating New Data Type

All the following criteria must be met:

- A new meaningful type of BCC can be found. The meaning should be clear and distinct from that of other types.
- It must be possible to assign to this type a unique Data Type term.
- The new term must reflect well recognized day-to-day data naming practices.
- It must be possible to clearly describe the usage of the term in contrast with other similar terms.
- Two CDT's must differ in structure: i.e. by simple vs. complex structure, by the value domain assigned to the Content and Supplementary Components.

Clarifications:

- If the new type has the same structure as an existing type but differs in the usage of the supplementary components or requires additional supplementary components, the new type can only be created provided that a meaningful Data Type term and usage can be found as described above, otherwise the existing type should be changed, preferably in a backward compatible way.
    - The value domain of a CDT must be unique. If you can derive the value domain from that of an existing CDT, you should define a BDT.
    - On the other hand unique semantics and usage can be taken into consideration.
- When synonymous terms exist (no difference is found in meaning, structure, usage, data naming practice), a preferred term must be chosen. The synonymous terms must be recorded as business terms.

C. 2 Criteria For Creating New Primitive Type

It is legitimate to create a new CCTS primitive type whenever a new reusable combination of value domain and representation can be identified, provided that this new primitive type is required in a CDT.

Possible cases:

- A new primitive value domain has been identified, i.e. defined axiomatically from fundamental notions, or as an outright enumeration.
    - Boolean is an outright enumeration as defined in ISO 11404.

  - Integer is defined axiomatically from fundamental notions.
  - Decimal or binary “floating point numbers” define a value domain (see IEEE 754) that is distinct from real numbers by adding special values such as infinite.
- A new derived value domain has been identified, which cannot be expressed by facet restrictions, and thus requires its own identity.
  - For example ‘URI’ is derived from the ‘string’ primitive value domain based on a referenceable specification rather than on facet restrictions: we cannot identify this value domain without a new named primitive.
  - Decimal is a subset of the set of real numbers, but is widely accepted as a primitive in its own right.
- Value domain that can be expressed by a facet restriction from an existing primitive, but that is so frequently widely adopted and supported that it deserves to be treated as a primitive in its own right.
  - For example, ‘token’ is a restriction of string using the whitespace facet, but it is widely accepted as a built-in type in standards such W3C xsd part 2.
  - Float and Double are restrictions of binary floating point but are widely accepted as standards such as IEEE 754.
- A new reusable representation of the primitive value domain has been identified.
  - For expressing date and time values we need a functional representation to support processing: this is what ISO 8601 specifies.
  - “Date” and “Time” are reusable representations of the date-and-time value domain.

C. 3 Data Maintenance Request Submission Procedures

Submissions must be made to the UN/CEFACT Entry Point – TBG16 – using the submission form found at:

http://www1.unece.org/cefact/platform/download/attachments/41582719/Data+Type+Catalogue+DMR+Form+Draft+B.doc?version=1

Appendix D Date. Type, Date Time. Type, and Time. Type Representations

D. 1 Allowed ISO 8601 Date. Type Representation

Representation Category	Name	Extended ISO 8601 representation
Calendar Date	Calendar Date	YYYY-MM-DD
Calendar Date with reduced precision	Year and Month	YYYY-MM
	Year	YYYY
	Century	YY
Truncated Calendar Date	Month and Day	--MM-DD
	Month	--MM--
	Day of Month	---DD
Ordinal Date	Ordinal Date	YYYY-DDD
Trucated Ordinal Date	Day of Year	-DDD
Week Date	Week Date	YYYY-Www-D
Week Date with reduced precision	Year and Week	YYYY-Www
Truncated Week Date	Week and Day of Week	-Www-D
	Week of Year	-Www
	Day of Week	-W-D

D. 2 Allowed ISO 8601 Time . Type Representation

Second decimals are allowed and optional. All types that include seconds also include second decimals.

UTC and UTC Offset time types are available for all types that include hours.

Representation Category	Name	Extended ISO 8601 representation
Time of Day	Time (local)	hh:mm:ss,s
	Time (UTC)	hh:mm:ss,sZ
	Time (with UTC offset)	hh:mm:ss,s+hh:mm
Time of Day with reduced precision	Hour and Minute (local)	hh:mm
	Hour and Minute (UTC)	hh:mmZ
	Hour and Minute (with UTC offset)	hh:mm+hh:mm
	Hour (local)	hh
	Hour (UTC)	hhZ
	Hour (with UTC offset)	hh+hh:mm
Truncated Time of Day	Minute and Second of Hour	-mm:ss,s
	Minute of Hour	-mm
	Second of Minute	--ss,s

D. 3 Allowed ISO 8601 Date Time . Type Representation

Second decimals are allowed and optional. All types that include seconds also include second decimals.

UTC and UTC Offset time types are available for all types that include hours.

Representation Category	Name	Extended ISO 8601 representation
Calendar Date and Time	Calendar Date and Time	YYYY-MM-DDThh:mm:ss
	Calendar Date and Time (with UTC offset)	YYYY-MM-DDThh:mm:ss+hh:mm
	Calendar Date and Time (UTC)	YYYY-MM-DDThh:mm:ssZ
Calendar Date and Time with reduced precision	Calendar Date, Hour and Minute	YYYY-MM-DDThh:mm
	Calendar Date, Hour and Minute (with UTC offset)	YYYY-MM-DDThh:mm+hh:mm
	Calendar Date, Hour and Minute (UTC)	YYYY-MM-DDThh:mmZ
	Calendar Date and Hour	YYYY-MM-DDThh
	Calendar Date and Hour (with UTC offset)	YYYY-MM-DDThh+hh:mm
	Calendar Date and Hour (UTC)	YYYY-MM-DDThhZ
	Calendar Date	YYYY-MM-DD
Truncated Calendar Date and Time	Month, Day and Time	--MM-DDThh:mm:ss
	Month, Day and Time (with UTC offset)	--MM-DDThh:mm:ss+hh:mm
	Month, Day and Time (UTC)	--MM-DDThh:mm:ssZ
Truncated Calendar Date and Time with reduced precision	Month, Day, Hour and Minute	--MM-DDThh:mm
	Month, Day, Hour and Minute (with UTC offset)	--MM-DDThh:mm+hh:mm
	Month, Day, Hour and Minute (UTC)	--MM-DDThh:mmZ
	Month, Day and Hour	--MM-DDThh
	Month, Day and Hour (with UTC offset)	--MM-DDThh+hh:mm
	Month, Day and Hour (UTC)	--MM-DDThhZ
	Month and Day	--MM-DD
Truncated Calendar Date and Time	Day of Month and Time	---DDThh:mm:ss

Representation Category	Name	Extended ISO 8601 representation
	Day of Month and Time (with UTC offset)	---DDThh:mm:ss+hh:mm
	Day of Month and Time (UTC)	---DDThh:mm:ssZ
Truncated Calendar Date and Time with reduced precision	Day of Month, Hour and Minute	---DDThh:mm
	Day of Month, Hour and Minute (with UTC offset)	---DDThh:mm+hh:mm
	Day of Month, Hour and Minute (UTC)	---DDThh:mmZ
	Day of Month and Hour	---DDThh
	Day of Month and Hour (with UTC offset)	---DDThh+hh:mm
	Day of Month and Hour (UTC)	---DDThhZ
	Day of Month	---DD
Ordinal Date and Time	Ordinal Date and Time	YYYY-DDDThh:mm:ss
	Ordinal Date and Time (with UTC offset)	YYYY-DDDThh:mm:ss+hh:mm
	Ordinal Date and Time (UTC)	YYYY-DDDThh:mm:ssZ
Ordinal Date and Time with reduced precision	Ordinal Date, Hour and Minute	YYYY-DDDThh:mm
	Ordinal Date, Hour and Minute (with UTC offset)	YYYY-DDDThh:mm+hh:mm
	Ordinal Date, Hour and Minute (UTC)	YYYY-DDDThh:mmZ
	Ordinal Date and Hour	YYYY-DDDThh
	Ordinal Date and Hour (with UTC offset)	YYYY-DDDThh+hh:mm
	Ordinal Date and Hour (UTC)	YYYY-DDDThhZ
	Ordinal Date	YYYY-DDD
Truncated Ordinal Date and Time	Day of Year and Time	-DDDThh:mm:ss
	Day of Year and Time (with UTC offset)	-DDDThh:mm:ss+hh:mm
	Day of Year and Time (UTC)	-DDDThh:mm:ssZ
Truncated Ordinal Date and Time with reduced precison	Day of Year, Hour and Minute	-DDDThh:mm
	Day of Year, Hour and Minute (with UTC offset)	-DDDThh:mm+hh:mm
	Day of Year, Hour and Minute (UTC)	-DDDThh:mmZ
	Day of Year and Hour	-DDDThh
	Day of Year and Hour (with UTC offset)	-DDDThh+hh:mm

Representation Category	Name	Extended ISO 8601 representation
	Day of Year and Hour (UTC)	-DDDThhZ
	Day of Year	-DDD
Week Date and Time	Week Date and Time	YYYY-Www-DThh:mm:ss
	Week Date and Time (with UTC offset)	YYYY-Www-DThh:mm:ss+hh:mm
	Week Date and Time (UTC)	YYYY-Www-DThh:mm:ssZ
Week Date and Time with reduced precision	Week Date, Hour and Minute	YYYY-Www-DThh:mm
	Week Date, Hour and Minute (with UTC offset)	YYYY-Www-DThh:mm+hh:mm
	Week Date, Hour and Minute (UTC)	YYYY-Www-DThh:mmZ
	Week Date and Hour	YYYY-Www-DThh
	Week Date and Hour (with UTC offset)	YYYY-Www-DThh+hh:mm
	Week Date and Hour (UTC)	YYYY-Www-DThhZ
	Week Date	YYYY-Www-D
Truncated Week Date and Time	Week and Day of Year and Time	-Www-DThh:mm:ss
	Week and Day of Year and Time (with UTC offset)	-Www-DThh:mm:ss+hh:mm
	Week and Day of Year and Time (UTC)	-Www-DThh:mm:ssZ
Truncated Week Date and Time with reduced precision	Week and Day of Year, Hour and Minute	-Www-DThh:mm
	Week and Day of Year, Hour and Minute (with UTC offset)	-Www-DThh:mm+hh:mm
	Week and Day of Year, Hour and Minute (UTC)	-Www-DThh:mmZ
	Week and Day of Year and Hour	-Www-DThh
	Week and Day of Year and Hour (with UTC offset)	-Www-DThh+hh:mm
	Week and Day of Year and Hour (UTC)	-Www-DThhZ
	Week and Day of Year	-Www-D
Truncated Week Date and Time	Day of Week and Time	-W-DThh:mm:ss
	Day of Week and Time (with UTC offset)	-W-DThh:mm:ss+hh:mm
	Day of Week and Time (UTC)	-W-DThh:mm:ssZ
Truncated Week Date and Time with reduced precision	Day of Week, Hour and Minute	-W-DThh:mm
	Day of Week, Hour and Minute (with UTC offset)	-W-DThh:mm+hh:mm

Representation Category	Name	Extended ISO 8601 representation
	Day of Week, Hour and Minute (UTC)	-W-DThh:mmZ
	Day of Week and Hour	-W-DThh
	Day of Week and Hour (with UTC offset)	-W-DThh+hh:mm
	Day of Week and Hour (UTC)	-W-DThhZ
	Day of Week	-W-D

D. 4 Excluded Representations

Representation Category	Extended ISO 8601 representation
Century truncated Calendar Dates	-YY
	-YY-MM
	YY-MM-DD
Century truncated Ordinal Dates	YY-DDD
Century truncated Week Dates	YY-Www
	YY-Www-D
Decade truncated Week Dates	-Y-Www
	-Y-Www-D
Reduced precision Time with decimal	hhmm,m
	hh,h
Truncated Time with UTC offset	-mmss+hh:mm
	-mm+hh:mm
	--ss+hh:mm
Truncated Time UTC	-mmssZ
	-mmZ
	--ssZ

Also excluded: any Date and Time representation that includes one of the above exclusions