V2\_SPEC\_DT\_FLAVORS\_R1\_D1\_2018MAY



**HL7 Version 2 Standardized Data Type Specializations   
Release 1**

May 2018

**HL7 Draft Ballot for Comments**

**Sponsored by:  
Conformance Work Group**

|  |  |
| --- | --- |
| Conformance Co-chair: Principle Author | Robert Snelick, National Institute of Standards and Technology (NIST) |
| Conformance Co-chair | Frank Oemig |
| Conformance Co-chair | Nathan Bunker, American Immunization Registry Association (AIRA) |
| Conformance Co-chair | Ioana Singureanu |
| Public Health Co-chair: Principle Author | Craig Newman, Centers for Disease Control and Prevention (CDC) |

**Questions or comments regarding this document should be directed to the Conformance Workgroup (**[cgit@lists.hl7.org](mailto:cgit@lists.hl7.org)**).**

Copyright © 2018 Health Level Seven International ® ALL RIGHTS RESERVED. The reproduction of this material in any form is strictly forbidden without the written permission of the publisher. HL7 and Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. Pat & TM Off**.**

Use of this material is governed by HL7's [**IP Compliance Policy**](http://www.hl7.org/legal/ippolicy.cfm?ref=nav).

Acknowledgments

The authors of this document wish to recognize the following participants who contributed their time and expertise to the development of this specification.

|  |  |
| --- | --- |
|  |  |
|  |  |
| Sheryl Taylor | National Institute of Standards and Technology (NIST) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table of Contents

[1 Introduction 9](#_Toc509483252)

[1.1 Background and Purpose 9](#_Toc509483253)

[1.1.1 Relevant Specifications 9](#_Toc509483254)

[1.1.2 Requisite Knowledge 10](#_Toc509483255)

[1.1.3 Definitions and Acronyms 10](#_Toc509483256)

[2 Definition and Methodology 11](#_Toc509483257)

[2.1 Use 13](#_Toc509483258)

[2.2 Scope 14](#_Toc509483259)

[2.3 Binding 15](#_Toc509483260)

[2.3.1 Compliance 16](#_Toc509483261)

[2.4 User Data Type Flavors 16](#_Toc509483262)

[2.5 Nested Data Type Flavors 17](#_Toc509483263)

[2.6 Handling of Versions 19](#_Toc509483264)

[3 Data Type Flavor Specification 20](#_Toc509483265)

[3.1 Data Type Flavor Definition Template 20](#_Toc509483266)

[3.2 Identifier Naming Rules and Conventions 21](#_Toc509483267)

[4 Standardized Data Type Flavor Library 24](#_Toc509483268)

[4.1 Process and Guidelines Creating SDTFs 24](#_Toc509483269)

[4.2 Data Types 25](#_Toc509483270)

[4.2.1 CWE – Coded with Exceptions 25](#_Toc509483271)

[4.2.1.1 CWE\_01 - Coded with Exceptions 25](#_Toc509483272)

[4.2.1.2 CWE\_02 - Coded with Exceptions 27](#_Toc509483273)

[4.2.1.3 CWE\_03 - Coded with Exceptions 29](#_Toc509483274)

[4.2.1.4 CWE\_04 - Coded with Exceptions 31](#_Toc509483275)

[4.2.1.5 CWE\_05 - Coded with Exceptions 33](#_Toc509483276)

[4.2.1.6 CWE\_06 - Coded with Exceptions 35](#_Toc509483277)

[4.2.1.7 CWE\_07 - Coded with Exceptions 36](#_Toc509483278)

[4.2.1.8 CWE\_08 - Coded with Exceptions 37](#_Toc509483279)

[4.2.2 CX - Extended Composite ID with Check Digit 38](#_Toc509483280)

[4.2.2.1 CX\_01 - Extended Composite ID with Check Digit 39](#_Toc509483281)

[4.2.2.2 CX\_02 - Extended Composite ID with Check Digit 40](#_Toc509483282)

[4.2.2.3 CX\_03 - Extended Composite ID with Check Digit 40](#_Toc509483283)

[4.2.2.4 CX\_04 - Extended Composite ID with Check Digit 41](#_Toc509483284)

[4.2.3 DTM – Date/time 42](#_Toc509483285)

[4.2.3.1 DTM\_01 - Date/Time 42](#_Toc509483286)

[4.2.3.2 DTM\_02 - Date/Time 43](#_Toc509483287)

[4.2.3.3 DTM\_03 - Date/Time 44](#_Toc509483288)

[4.2.3.4 DTM\_05 - Date/Time 45](#_Toc509483289)

[4.2.3.5 DTM\_06 - Date/Time 46](#_Toc509483290)

[4.2.3.6 DTM\_07 - Date/Time 46](#_Toc509483291)

[4.2.3.7 DTM\_08 - Date/Time 47](#_Toc509483292)

[4.2.3.8 DTM\_09 - Date/Time 48](#_Toc509483293)

[4.2.3.9 DTM\_10 - Date/Time 49](#_Toc509483294)

[4.2.3.10 DTM\_11 - Date/Time 50](#_Toc509483295)

[4.2.3.11 DTM\_12 - Date/Time 50](#_Toc509483296)

[4.2.3.12 DTM\_13 - Date/Time 51](#_Toc509483297)

[4.2.3.13 DTM\_14 - Date/Time 52](#_Toc509483298)

[4.2.4 ED - Encapsulated Data 53](#_Toc509483299)

[4.2.4.1 ED\_01 - Encapsulated Data 53](#_Toc509483300)

[4.2.5 EI - Entity Identifier 54](#_Toc509483301)

[4.2.5.1 EI\_01 - Entity Identifier 54](#_Toc509483302)

[4.2.5.2 EI\_02 - Entity Identifier 55](#_Toc509483303)

[4.2.5.3 EI\_03 - Entity Identifier 56](#_Toc509483304)

[4.2.5.4 EI\_04 - Entity Identifier 56](#_Toc509483305)

[4.2.5.5 EI\_05 - Entity Identifier 57](#_Toc509483306)

[4.2.5.6 EI\_06 - Entity Identifier 57](#_Toc509483307)

[4.2.6 EIP - Entity Identifier Pair 58](#_Toc509483308)

[4.2.6.1 EIP\_01 - Entity Identifier Pair 58](#_Toc509483309)

[4.2.6.2 EIP\_02 - Entity Identifier Pair 59](#_Toc509483310)

[4.2.6.3 EIP\_03 - Entity Identifier Pair 60](#_Toc509483311)

[4.2.6.4 EIP\_04 - Entity Identifier Pair 60](#_Toc509483312)

[4.2.6.5 EIP\_05 - Entity Identifier Pair 61](#_Toc509483313)

[4.2.7 ERL - Error Location 61](#_Toc509483314)

[4.2.7.1 ERL\_01 - Error Location 61](#_Toc509483315)

[4.2.7.2 ERL\_02 - Error Location 62](#_Toc509483316)

[4.2.8 FN - Family Name 64](#_Toc509483317)

[4.2.8.1 FN\_01 - Family Name 64](#_Toc509483318)

[4.2.8.2 FN\_02 - Family Name 64](#_Toc509483319)

[4.2.9 HD - Hierarchic Designator 65](#_Toc509483320)

[4.2.9.1 HD\_01 - Hierarchic Designator 65](#_Toc509483321)

[4.2.9.2 HD\_02 - Hierarchic Designator 66](#_Toc509483322)

[4.2.9.3 HD\_03 - Hierarchic Designator 67](#_Toc509483323)

[4.2.9.4 HD\_04 - Hierarchic Designator 67](#_Toc509483324)

[4.2.9.5 HD\_05 - Hierarchic Designator 68](#_Toc509483325)

[4.2.9.6 HD\_06 - Hierarchic Designator 69](#_Toc509483326)

[4.2.10 MSG - Message Type 70](#_Toc509483327)

[4.2.10.1 MSG\_01 - Message Type 70](#_Toc509483328)

[4.2.10.2 MSG\_02 - Message Type 70](#_Toc509483329)

[4.2.11 OG - Observation Grouper 71](#_Toc509483330)

[4.2.11.1 OG\_01 - Observation Grouper 71](#_Toc509483331)

[4.2.11.2 OG\_02 - Observation Grouper 71](#_Toc509483332)

[4.2.12 PT - Processing Type 72](#_Toc509483333)

[4.2.12.1 PT\_01 - Processing Type 72](#_Toc509483334)

[4.2.12.2 PT\_02 - Processing Type 72](#_Toc509483335)

[4.2.13 SAD - Street Address 73](#_Toc509483336)

[4.2.13.1 SAD\_01 - Street Address 73](#_Toc509483337)

[4.2.13.2 SAD\_02 - Street Address 73](#_Toc509483338)

[4.2.14 SN - Structured Numeric 74](#_Toc509483339)

[4.2.14.1 SN\_01 - Structured Numeric 74](#_Toc509483340)

[4.2.15 VID - Version Identifier 75](#_Toc509483341)

[4.2.15.1 VID\_01 - Version Identifier 75](#_Toc509483342)

[4.2.15.2 VID\_02 - Version Identifier 75](#_Toc509483343)

[4.2.15.3 VID\_03 - Version Identifier 76](#_Toc509483344)

[4.2.16 XAD - Extended Address 76](#_Toc509483345)

[4.2.16.1 XAD\_01 - Extended Address 77](#_Toc509483346)

[4.2.16.2 XAD\_02 - Extended Address 77](#_Toc509483347)

[4.2.16.3 XAD\_03 - Extended Address 78](#_Toc509483348)

[4.2.16.4 XAD\_04 - Extended Address 79](#_Toc509483349)

[4.2.16.5 XAD\_05 - Extended Address 80](#_Toc509483350)

[4.2.16.6 XAD\_06 - Extended Address 82](#_Toc509483351)

[4.2.16.7 XAD\_07 - Extended Address 83](#_Toc509483352)

[4.2.17 XCN - Extended Composite ID Number and Name for Persons 85](#_Toc509483353)

[4.2.17.1 XCN\_01 - Extended Composite ID Number and Name for Persons 85](#_Toc509483354)

[4.2.17.2 XCN\_02 - Extended Composite ID Number and Name for Persons 87](#_Toc509483355)

[4.2.17.3 XCN\_03 - Extended Composite ID Number and Name for Persons 88](#_Toc509483356)

[4.2.17.4 XCN\_04 - Extended Composite ID Number and Name for Persons 90](#_Toc509483357)

[4.2.18 XON - Extended Composite Name and Identification Number for Organizations 92](#_Toc509483358)

[4.2.18.1 XON\_01 - Extended Composite Name and Identification Number for Organizations 92](#_Toc509483359)

[4.2.18.2 XON\_02 - Extended Composite Name and Identification Number for Organizations 93](#_Toc509483360)

[4.2.18.3 XON\_03 - Extended Composite Name and Identification Number for Organizations 94](#_Toc509483361)

[4.2.18.4 XON\_04 - Extended Composite Name and Identification Number for Organizations 96](#_Toc509483362)

[4.2.18.5 XON\_05 - Extended Composite Name and Identification Number for Organizations 96](#_Toc509483363)

[4.2.19 XPN - Extended Person Name 97](#_Toc509483364)

[4.2.19.1 XPN\_01 - Extended Person Name 98](#_Toc509483365)

[4.2.19.2 XPN\_02 - Extended Person Name 99](#_Toc509483366)

[4.2.19.3 XPN\_03 - Extended Person Name 100](#_Toc509483367)

[4.2.19.4 XPN\_04 - Extended Person Name 101](#_Toc509483368)

[4.2.19.5 XPN\_05 - Extended Person Name 102](#_Toc509483369)

[4.2.20 XTN - Extended Telecommunication Number 103](#_Toc509483370)

[4.2.20.1 XTN\_01 - Extended Telecommunication Number 104](#_Toc509483371)

[4.2.20.2 XTN\_02 - Extended Telecommunication Number 105](#_Toc509483372)

[5 Appendix A – Base Standard Data Type Evolution 107](#_Toc509483373)

[6 Glossary 111](#_Toc509483374)

Index of Tables

[Table 1: Base CE (Coded Element) Data Type Definition 11](#_Toc509483375)

[Table 2: CE\_01 (Coded Element) Data Type Definition 12](#_Toc509483376)

[Table 3: CE\_02 (Coded Element) Data Type Definition 13](#_Toc509483377)

[Table 4: Using DTFs in Message Profiles 15](#_Toc509483378)

[Table 5: Added Constraint at Point-of-Use 15](#_Toc509483379)

[Table 6: Nested Data Type Flavors 18](#_Toc509483380)

[Table 7: HD\_01 Data Type Definition 18](#_Toc509483381)

[Table 8: DTF Construction Progression and Identifier Naming Rules 22](#_Toc509483382)

[Table 9: Character Key 22](#_Toc509483383)

[Table 10: Base Standard Data Type Evolution 107](#_Toc509483384)

Index of Figures

**No table of figures entries found.**

# Introduction

Data types and specializations of data types are core aspects of the HL7 v2.x standard and implementation guides. The HL7 v2.x standard provides a set of data types (referred to as the “base” data types). Typically, the base data types are not used “as is” in implementation guides; instead they are constrained for a particular need in the profiling process. The specialization of the data type is referred to as a data type “flavor”. A “standardized” set of data type flavors is defined in this specification. The intent of standardized data type flavors (SDTF) is to promote consistency and reuse of commonly used data type specializations.

This specification provides a definition of a data type flavor, a methodology for creating data type flavors, and a library of standardized data type flavor definitions that HL7 v2.x implementation guide authors can use. The library provides “off-the-shelf” pre-defined data type flavors that have been constrained for a narrower purpose. Upon determining the requirements for an element, a selection of a suitable data type flavor can be applied.

## Background and Purpose

When HL7 v2.x implementation guides are developed, data types often are (and should be) constrained to suit a particular use for a message element. Therefore, multiple specializations of a base data type are typically needed, and each specialization necessitates a distinct specification and identifier. Each specialization becomes a data type flavor. Historically, data type flavors have been created independently in implementation guides that use their own style and naming conventions. Because of lack of coordination among the v2 authors, duplicate data type flavor definitions (in terms of requirements) are created with inconsistent naming conventions. This approach is problematic since the same requirements are being specified but with different documentation. This specification provides a repository (library) of reusable documented data type flavors with consistent specification and naming conventions. The objectives and goals of standardizing data type flavors include providing:

* A methodology for creating standardized data type flavors
* A library of standardized data type flavors
* Guidance on use of the data type flavors

This document does not attempt to provide an exhaustive list of possible data type flavors. Only a core set of common data types with common specialization of those data types is defined, anticipating that the set will increase over time.

### Relevant Specifications

The HL7 Version 2 standardized data type flavor library is intended to be used in the creation of implementation guides and message profiles for the following standards:

* HL7 v2.3
* HL7 v2.3.1
* HL7 v2.4
* HL7 v2.5
* HL7 v2.5.1
* HL7 v2.6
* HL7 v2.7
* HL7 v2.7.1
* HL7 v2.8
* HL7 v2.8.1
* HL7 v2.8.2
* HL7 v2.9
* Any future HL7 v2 standard releases (e.g., HL7 v2.+)

### Requisite Knowledge

* HL7 v2.x Messaging Standard ([www.hl7.org](http://www.hl7.org))

### Definitions and Acronyms

This specification uses several important definitions and acronyms that the reader needs to be familiar with for better understanding. A brief overview is given here and further explanation is given in the remaining of the text.

**Data Type Flavor:** Is a specialization of a base data type flavor. That is, a data type defined in the base HL7 v2.x standard is constrained for a particular use, that constrained data type is given an identifier and is deemed a data type flavor.

**SDTF (Standard Data Type Flavor):** Is a Data Type Flavor created and managed by HL7 and is the central focus of this document.

**UDTF (User Data Type Flavor):** Is a Data Type Flavor created and managed by the user of the HL7 v2.x standard. This document provides creation and management guidelines and naming conventions for UDTFs.

**DTF – Data Type Flavor:** Is the generic term to include both SDTF (Standard Data Type Flavor) and UDTF (User Data Type Flavor).

# Definition and Methodology

A data type flavor is a specialization on a base data type. A specialization is created in the profiling process following the constraint rules published in the HL7 v2.x conformance methodology. The base data type definition provides a framework for a data structure in which, generally, most child elements (components) are optional. Table 1 illustrates the base data type definition (abbreviated) for the Coded Element (CE) data type in HL7 v2.5.1.

Table : Base CE (Coded Element) Data Type Definition

| CE (CODED ELEMENT) DATA TYPE | | | |
| --- | --- | --- | --- |
| SEQ | Name | DT | Usage |
| 1 | Identifier | ST | O |
| 2 | Text | ST | O |
| 3 | Name of Coding System | ID | O |
| 4 | Alternate Identifier | ST | O |
| 5 | Alternate Text | ST | O |
| 6 | Name of Alternate Coding System | ID | O |

In short, the data type is used to convey up to two codes for a data element. The data type is implicitly separated into two triplets. In the first triplet (components 1, 2, and 3) component 1 indicates the identifier (or the code), component 2 indicates the description for the code, and component 3 indicates the code system from which the code (component 1) is drawn. Similarly, the second triplet (components 4, 5, and 6) provides functionality for alternate coding. As shown in the definition, all the components in the base standard are optional (O).

In practice, the CE data type is constrained depending on the requirements needed for the element to which the data type is bound. For example, PID-10 (Race) is assigned the CE data type. Depending on requirements, the implementation guide author will constrain the data type as needed.

Previous methods included defining the data type constraints “in-line”, e.g., expanding the PID-10 element to include all components and constrain directly. The disadvantages with this method are that the definition is not reusable, and it also is hard to manage changes when a common specialization is used frequently in a specification. Another common approach is to conflate the requirements for the data type (i.e., either over or under specify the requirements for the data type use throughout the entire specification). This method is flawed because it does not specify the requirements sufficiently, and requirements can’t be disambiguated at the level of the element where the data type is used. A third method is to create and document data type flavors in an ad-hoc fashion. This approach does not promote reuse nor enable consistency across HL7 v2 specifications.

Table 2 shows the CE\_01 data type flavor in which the base CE data type is constrained such that a code and its constitute parts are required. In the example shown, component 1 is required (R), i.e., a code is required. Component 2 is required, i.e., the description for the code must be given. Component 3 also is required (R); given that a code is required, the code system from which the code is drawn must be present. The data type flavor is assigned an identifier; the identifier is the “root” data type identifier (CE), a delimiter “\_”, and an extension “\_01”. The CE\_01 data type flavor is conveying that the first triplet is required and the second triplet is optional. An appropriate short description for CE\_01 could be “Coded Element – First Triplet Required”.

Table : CE\_01 (Coded Element) Data Type Definition

| CE\_01 (CODED ELEMENT) DATA TYPE | | | |
| --- | --- | --- | --- |
| SEQ | Name | DT | Usage |
| 1 | Identifier | ST | R |
| 2 | Text | ST | R |
| 3 | Name of Coding System | ID | R |
| 4 | Alternate Identifier | ST | O |
| 5 | Alternate Text | ST | O |
| 6 | Name of Alternate Coding System | ID | O |

Table 3 shows the CE\_02 data type flavor in which the base CE data type is constrained such that a code and its constitute parts are required for the first triplet, and a code and its constitute parts must be supported for the second triplet. In the example shown, component 1 is required (R), i.e., a code is required. Component 2 is required, i.e., the description for the code must be given. Component 3 also is required (R); given that a code is required, the code system from which the code is drawn must be present. Component 4 is required, but may be empty (RE), i.e., the ability to provide an alternate code must be supported, but it may or may not appear in a given message instance. Component 5 and component 6 are conditionally required (R); i.e., if component 4 is valued, then components 5 and 6 must be valued. If component 4 is not valued, then components 5 and 6 must not be valued. Regardless of component 4’s state of presence, support for components 5 and 6 must be implemented. CE\_02 is specifying that the first triplet must be supported and valued and that the second triplet must be supported and may be valued. An appropriate short description for CE\_01 could be “Coded Element – First Triplet Required, Second Triplet Supported”.

Table : CE\_02 (Coded Element) Data Type Definition

| CE\_02 (CODED ELEMENT) DATA TYPE | | | | |
| --- | --- | --- | --- | --- |
| SEQ | Name | DT | Usage | Condition Predicate |
| 1 | Identifier | ST | R |  |
| 2 | Text | ST | R |  |
| 3 | Name of Coding System | ID | R |  |
| 4 | Alternate Identifier | ST | RE |  |
| 5 | Alternate Text | ST | C(R/X) | If CE.4 is valued. |
| 6 | Name of Alternate Coding System | ID | C(R/X) | If CE.4 is valued. |

Note that, since the usage for components 5 and 6 are conditionally based on the content of component 4, explicit condition predicates are specified in the data type flavor. Explicit condition predicates are not given in the base standard and historically[[1]](#footnote-1) are not given in implementation guides. In the creation of the data type flavor library, a point of emphasis is to give explicit requirements including the condition predicates. Additionally, a shorthand notation is used for identifying the data type in the condition predicate. If using the expanded identifier, the condition predicate would be expressed as “If CE\_02.4 is valued.”. The context in which the condition predicate is defined is used and provides the knowledge to resolve the true identifier. The shorthand notation is specified for improved readability.

## Use

Figure 1 provides an overview of the process for creation and use of SDTF and refinements to those flavors for use in implementation guides. The HL7 v2 standard defined a set of base data types that is “pulled in” into implementation guides as a starter set of data types. These data types are used “as is” or constrained for a particular use. For example, if a field element is kept as optional as defined in the base standard, then the associated data type is likely to be used “as is”. That is, no constraints are added to the element. If, however, the element is made required, then the data type is likely to be constrained to satisfy the intended use. In this case, a SDTF should be used when possible.

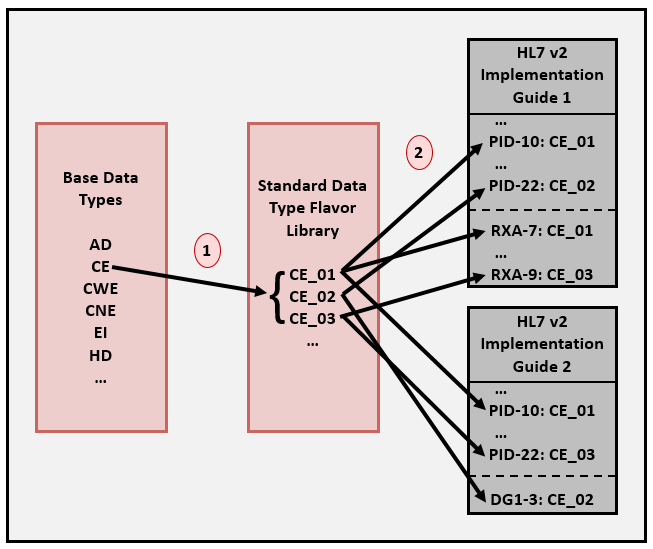


Figure : Creation and Use of Standardized Data Type Flavors

As shown in Figure 1, the HL7 v2 standards have a set of base data types (a sample set is indicated in the box labelled “Base Data Types”). From these base data types, zero or more SDTFs are created. The box labelled “Standard Data Type Flavor Library” illustrates that three flavors are created for the CE data type.

The rightmost boxes, labelled “HL7 v2 Implementation Guide 1 and 2”, illustrates examples of how the SDTFs could be used in implementation guides. In the first implementation guide: for PID-10, the base data type of CE is replaced with the SDTF CE\_01 in the PID segment definition table; SDTF CE\_02 is used for PID-22; CE\_01 is used again for RXA-7 in the RXA segment definition table; and RXA-9 uses the CE\_03 STDF. The diagram also shows the use of SDTFs in another implementation guide, thus emphasizing the benefit of reuse and consistency.

## Scope

The scope of a SDTF specialization is generally limited to the usage indicator, therefore, a SDTF is not a complete specialization of the data type. Certain attributes, such as the vocabulary bindings, need to be defined at the point-of-use. This delineation is both necessary and practical. A goal of this project is to define a commonly used set of specialization patterns at the level that allows for reusable components. Otherwise, the set of possible combinations would limit utility. Further specialization of a SDTF for specific use in a given context can be specified and managed in User Data Type Flavors (UDTF). Without the need for broad utility beyond that specific use, however, creation of a UDTF is not recommended. See Section 2.4 (User Data Type Flavors) for details and use.

DTFs are only defined for complex data types. Creating flavors for primitive data types has limited utility, and constraints for such data types are most applicable at the point-of-use. DTFs for primitive data types are out of scope and are not discussed further in this specification.

## Binding

Table 4: Using DTFs in Message Profiles shows examples of how a DTF is bound to a message element in a message profile segment definition. In the base standard for PID-10 (Race) the data type definition is “CE”, and in the process of profiling PID-10 is now assigned to “CE\_01”. Because of this profiling, the requirements for the PID-10 (Race) field are defined in the “CE\_01” data type. Likewise, a data type substitution of “XAD\_03” for “XAD” for the PID-11 (Patient Address) field occurred in the profiling process.

Table : Using DTFs in Message Profiles

| Patient Identification (PID) Segment Definition Excerpt | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| SEQ | Name | DT | Usage | Card. | Len | Vocabulary |
| … | … | … | … |  |  | … |
| 8 | Administrative Sex | IS | R |  | 1..1 | HL70001\_01 |
| 9 | Patient Alias |  | X |  |  |  |
| 10 | Race | CE\_01 | RE | 0..1 |  | HL70005\_01 |
| 11 | Patient Address | XAD\_03 | R | 1..3 |  |  |
| … | … | … | … |  |  | … |

Table 5 shows how additional constraints, such as length and vocabulary, can be added to the data type flavor at the point-of-use. The “CE\_01” flavor is bound to PID-10 (Race), and changes to that specific instance are made at that location for length and vocabulary. Note, only elements that are marked as required are profiled here. A minimum and maximum length is given for the first “CE\_01” triplet, and a specific version of the HL70396\_01 is bound to component 3.

Table : Added Constraint at Point-of-Use

| CE\_01 (CODED ELEMENT) DATA TYPE | | | | | |
| --- | --- | --- | --- | --- | --- |
| SEQ | Name | DT | Usage | Len | Vocabulary |
| 1 | Identifier | ST | R | 1..12 |  |
| 2 | Text | ST | R | 1..60 |  |
| 3 | Name of Coding System | ID | R | 1..20 | HL70396\_01 |
| 4 | Alternate Identifier | ST | O | 20 |  |
| 5 | Alternate Text | ST | O | 199 |  |
| 6 | Name of Alternate Coding System | ID | O | 20 | HL70396 |

For the optional elements (components 4, 5, and 6), both the length and vocabulary remain as they were defined in the base standard. A few observations are worth noting. First, the value set to be used for component 1 is defined at the field level (as indicated in Table 4. A decision by the specifier in this case was made to bind the value set of HL70396\_01 to component 3 at the point of use. An alternative could be to define the value set directly in a user data type flavor (See Section 2.4), which would apply to all uses of the CE\_01 DTF.

### Compliance

Table 4 presented examples of data type substitutions of a SDTF for a base data type definition. Profiling, however, often is a multistep process; therefore, a SDTF itself could be substituted for another SDTF, UDTF, or in-line constraints. In such cases, compliance rules must be adhered to, as they are for any other conformance construct. A data type substitution is compliant if all its parts are compliant. For example, if usages of HD\_01 components are R, R, and R, a data type substitution with HD\_02 where the components’ usages are RE, R, and R respectively would be deemed non-compliant. On the other hand, if HD\_01 was replacing HD\_02 in a profile definition, then that substitution is compliant since the data type is being further constrained not relaxed.

## User Data Type Flavors

A SDTF and specification completion at the point-of-use often is sufficient; however, there may be circumstances where a needed SDTF does not exist or there is a compelling reason to further specify the standardized data type flavor to create a derived data type flavor. These data type flavors are classified as User Data Type Flavors (UDTF). UDTF can be useful if it is more efficient to create highly specialized flavors in a set of implementation guides managed by a specific user or set of users; but highly specialized local flavors have limited reuse utility.

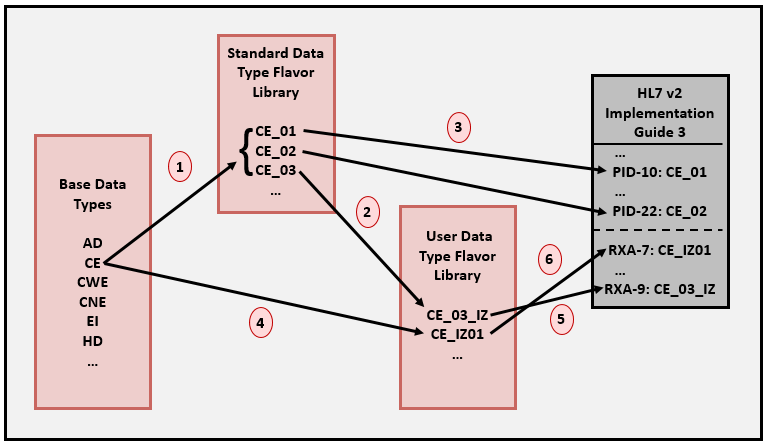


Figure : Creation and Use of SDTF and UDTF

Figure 2 shows various paths that can be taken for the creation and use of both SDTFs and UDTFs. Point (1) shows the path for creating the set of SDTFs documented in a library (Section 4 documents this library). Point (3) shows the use of the SDTFs. In the implementation guide, further refinement can be specified in the context of its use for the element to which it is bound. Point (2) indicates the path where the SDTF is further specified and documented as a UDTF. Point (5) shows its use in an implementation guide. Point (4) shows the path when a UDTF is created directly from the base standard. This is a circumstance where a template flavor is needed but does not exist in the SDTF library. If the need for such a flavor is common, then the flavor should be proposed to be standardized. Point (6) shows the use of the UDTF in an implementation guide.

Like the SDTFs, the UDTFs have standardized naming requirements and conventions. Section 3.2 provides the details.

## Nested Data Type Flavors

Specification and examples shown to this point have focused on DTFs in which all the components making up the data type have primitive data types. Many data types, however, have complex data types at the component level (i.e., the data type has sub-components). These nested complex data types are supported in the same manner as the “flat” data types. Since there are DTFs within a DTF, the potential combinations can multiply quickly. Therefore, the SDTF library limits instances to the most common uses. Further specification is always available at the point-of-use, i.e., a DTF can be specified within another DTF directly in the implementation guide or localized UDTFs can be created.

Table : Nested Data Type Flavors

| CX\_01 (Extended Composite ID) DATA TYPE | | | |
| --- | --- | --- | --- |
| SEQ | Name | DT | Usage |
| 1 | ID Number | ST | R |
| 2 | Check Digit | ST | O |
| 3 | Check Digit Scheme | ID | O |
| 4 | Assigning Authority | HD\_01 | R |
| 5 | Identifier Type Code | ID | R |
| 6 | Assigning Facility | HD | O |
| 7 | Effective Date | DT | O |
| 8 | Expiration Date | DT | O |
| 9 | Assigning Jurisdiction | CWE | O |
| 10 | Assigning Agency or Department | CWE | O |

Table 6 and Table 7 illustrate an example of a nested SDTF. The CX\_01 SDTF uses the HD\_01 SDTF as part of its definition. HD\_01 shows how a DTF definition can also include explicit constraints in the form of conformance statements.

Table : HD\_01 Data Type Definition

| HD\_01 (Hierarchic Designator) DATA TYPE | | | |
| --- | --- | --- | --- |
| SEQ | Name | DT | Usage |
| 1 | Namespace ID | IS | RE |
| 2 | Universal ID | ST | R |
| 3 | Universal ID Type | ID | R |

| HD\_01 Conformance Statements | |
| --- | --- |
| ID | Description |
| HD\_01-01 | The value of HD.2 (Universal ID) SHALL be formatted with an ISO-compliant OID. |
| HD\_01-02 | The value of HD.3 (Universal ID Type) SHALL be 'ISO'. |

A “Purpose and Use” description can be the following:

*“Use of the CX\_01 flavor is recommended when the ID must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by binding the Assigning Authority (CX.4) to the HD\_01 flavor which requires the Universal ID (HD.2) and constraining it to be an ISO complaint OID.”*

For details of how to reference message elements, construction and identification of conformance statements, see the HL7 v2.x Conformance Methodology specification.

## Handling of Versions

The base HL7 v2 data types have changed over the various versions of the standard. The changes have included:

* Adding new data types
* Deprecating data types
* Redefining existing data types (e.g., CWE added components in version 2.7).

Modifications not only included changes in the structured definition, but also could change or clarify the semantic definition. The SDTFs are defined in alignment with the structure and definition of the underlying data type and are not tied to versions of the standard. Therefore, a SDTF can apply to more than one version of the standard. The criteria for determining a change in the base data type are as follows:

* number of components
* name of component
* data type of a component
* usage

The length and the vocabulary binding are not considered, even though they may have changed since these constructs are specified at the point-of-use. When using the SDTF, the initial length and vocabulary will be that of the base standard, and refinement of those constructs can be applied during the process of profiling.

Appendix A – Base Standard Data Type Evolution, provides a table that shows a list of all of the base data types for each HL7 v2.x version and also where a change occurred based on the criteria given above.

# Data Type Flavor Specification

The DTF specification consists of a definition and identifier naming rules and conventions. Section 3.1 presents the data type definition that includes narrative text, meta-data, and the data type constraints. Section 3.2 defines the rules for naming data type flavors. Section 3.3 discusses how the DTF definitions are defined with relationship to the HL7 v2.x versions.

## Data Type Flavor Definition Template

Figure 3 shows an example entry into a data type flavor library for “CE\_02” data type definition. The template indicates the name of the data type flavor, meta-data, the data type definition, and narrative text that further defines intent. Such text will include a “Purpose and Use” and, optionally, additional comments.

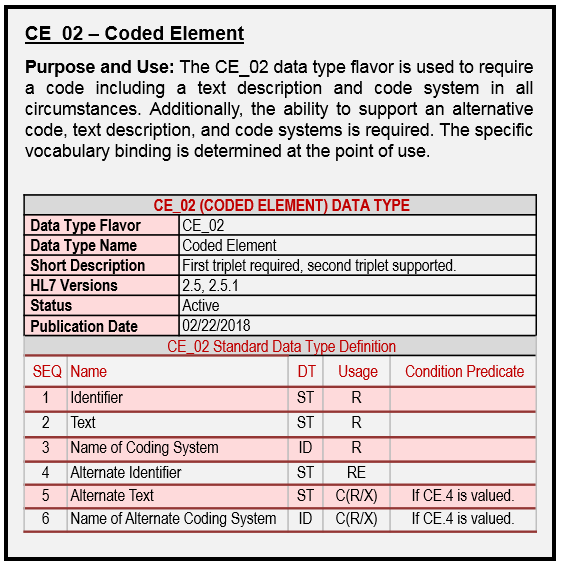
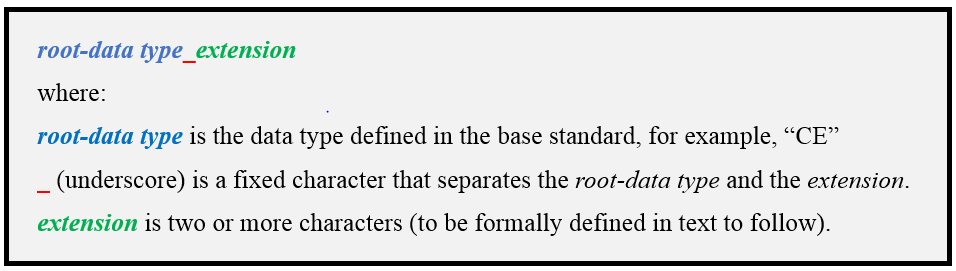


Figure : Example Template of DTF Definition

The meta-data include the Data Type Flavor Identifier, Root Data Type Name, and Short Description, which are self-explanatory. HL7 Versions refers to the HL7 v2 versions to which the data type flavor is applicable. In this example, the “CE\_02” is compatible with versions 2.5 and 2.5.1. Status can be either active or deprecated. Publication Date indicates the date the DTF became active and, when the DTF is deprecated, the Publication Date indicates the date range when it was active. The latter date in the range gives the date it was deprecated.

## Identifier Naming Rules and Conventions

All data type flavor definitions follow a set of naming rules and conventions. The basic grammar is as follows:



There are different naming rules and conventions for SDTFs and UDTFs.

Examples include:

CE\_01

CE\_01\_IZ

CE\_01\_IZ\_MI

CE\_02

CE\_VR01

CE\_VR01\_CT

HD\_01

HD\_02

HD\_03

Standardized data type flavor identifiers start with an extension of “01”, and for each successive flavor of the data type the identifier extension will increase by one, e.g., “02”, “03”, irrespective of the version of the standard to which the flavor is applicable. Flavors can be deprecated over time, in which case, the identifier will not be reused. There is no meaning associated with the identifier, other than the sequence in which it was created. The SDTF identifier provides no information about the HL7 v2 version to which it is applicable; the SDTF definition meta data contains that information.

The underlying structure of the data type does not change, so implementers can rely on the root data type always to determine the structured definition. Table 8 indicates the progression of DTF construction and identifier naming rules and conventions. The SDTF is a concatenation of the root data type identifier, an “\_”, and a two-digit extension (e.g., HD\_01). Table 8 shows the allowed patterns; a key is provided in Table 9 that indicates how the placeholders are interpreted. For example, ‘D’ is a digit [0..9]. SDTF definitions can progress to more refined UDTF definitions. The left-hand side of Table 8 indicates two UDTF levels. In the first UDTF level (HD\_DD\_AA) the extension adds the “\_AA” pattern. In the second level (HD\_DD\_AAC\*) any number of additional characters (indicated by “C\*”) can be added (this may include additional levels, although it is not anticipated that levels beyond the first UDTF level will be used often). The example shown for the two UDTF levels might indicate a national profile requirement at the first level, and a more refined requirement at a US state level, for example for an immunization registry use case.

The right-hand columns of Table 8 indicate the progression path for creating a UDTF directly from the base standard data type. SDTF should always be used, but in cases where a DTF is needed and it is not defined as a SDTF, a UDTF can be created. If such use is common, then the UDTF should be proposed as a SDTF.

Table : DTF Construction Progression and Identifier Naming Rules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Base to Standard Flavor Path** | | **Base to User Flavor Path** | |
|  | **Pattern** | **Example** | **Pattern** | **Example** |
| **HL7 Base** | HD | HD | HD | HD |
| **HL7 Standard Flavor** | HD\_DD | HD\_01 |  |  |
| **User**  **Flavor** | HD\_DD\_AA | HD\_01\_IZ | HD\_L\*DD | HD\_VR01 |
| **Derived User Flavor** | HD\_DD\_AAC\* | HD\_01\_IZ\_CT | HD\_L\*DDC\* | HD\_VR01\_CT |

Note that the rules for identifying UDTF definitions are different than those of SDTF definitions. SDTF definitions reserve an identifier space that can’t be used except for SDTF definitions.

Table : Character Key

|  |  |
| --- | --- |
| **Character** | **Definition** |
| **\_** | \_ |
| **D** | [0..9] |
| **A** | [0..9, A-Z] |
| **C** | [0..9, A-Z, \_] |
| **\*** | 1 to many |

As mentioned, along with the naming rules, certain conventions are followed when creating identifiers for the SDTFs. SDTFs are assigned a two-digit extension starting at “01” that is incremented for each subsequent definition. The assignment is arbitrary in the sense that no meaning is associated with the identifier and the identifier is not related to a version of the standard. These rules will be adhered to in the creation and on-going management of the SDTF library. For UDTFs, users should adhere to the conventions given in this specification.

# Standardized Data Type Flavor Library

Notes for ballot reviewers:

1. This specification provides a first draft of the proposed SDTF library. This initial set is not a complete list but rather a sample list for which discussion and refinements can be made. Further refinements are expected through the multiple ballot cycles. Initial versions of the library will be relatively small with the intent of adding flavors with each cycle. Initial focus is on the definition of the flavors and the mechanics of how to use them. Subsequent discussion will focus on the content of individual SDTF definitions.
2. The format of the SDTF definitions in this section does not match the format presented in Section 3.1 of this specification. This format is expected for the September 2018 ballot.
3. The National Institute of Standards and Technology (NIST) Implementation Guide Authoring and Management Tool (IGAMT) was used to create, maintain, and publish the SDTF Library. Future IGAMT export capabilities will include a web-based publication of the SDTF library.

## Process and Guidelines Creating SDTFs

A process and a set of guidelines were adopted when creating the SDTFs and are given in the list below:

* Concerning Usages, Condition Predicates, and Conformance Statements, at least one of these elements must be valued in each data type flavor. That is, it should not be possible to construct a scenario where the flavor allows all of these elements to be empty.
* When the base standard Optionality for an element is conditional (C) but the flavor intends for the Usage of the element to be silent, the flavor should be assigned a Usage of O to simplify the flavor. As with other optional elements, implementers who wish to support the element are referred to the base standard for requirements. One specific rule is that a Condition Predicate should be not be based on an optional element. For example, in the CWE data type, if CWE.4 (Alternate Identifier) is given a Usage of O, then CWE.6 should also be given a Usage of O even though the base standard Optionality is C.
* Possible alternatives to consider as part of the ballot process:
  + Rather than O, CWE.6 would be given a Usage of C without a Condition Predicate
  + Rather than O, CWE.6 would be given a fully defined Usage of C, including True and False Usages and a Condition Predicate
* For a SDTF, all Conformance Statements and Condition Predicates should be machine testable (i.e. no free text IGAMT statements).
* There is no need to create data type flavors for atomic data types (e.g., IS, ID, TX, etc).
* When a data type includes a non-optional (a Usage of R, RE or C) component that uses a complex data type, the complex data type should itself be a flavor and not reference the base standard.
  + Optional components using a complex data type may reference the base standard data type.
  + Components using an atomic data type should reference the base standard data type.
  + For example, CX.4 uses the HD data type. A CX data type flavor which requires support for CX.4 should specify an HD data type flavor (not the base HD data type) in its definition.
* When constructing Condition Predicates, the more restrictive Usage should be the True usage. For example, in the CWE data type, if the Original Text (CWE.9) is required when an Identifier (CWE.1) is unavailable but it is optional otherwise, the construct should be “C(R/O) if CWE.1 is not populated” rather than “C(O/R) if CWE.1 is populated”.
  + When the alternatives are R and X, then R should be the True Usage and X the False Usage
* When authoring an implementation guide, select the simplest data type that meets the use case requirements. For example, do not use a data type flavor that requires support for the globally unique IDs if it is not required by the IG use case.
* Do not assign a Usage of X to data type components unless it’s important that they not be sent. If a component is not critical to the data type, assign a Usage of O to allow trading partners to agree to exchange the data if necessary locally.
  + One exception is where the base standard has an Optionality of B (Backwards Compatible) or W (Withdrawn). In these scenarios, a Usage of X is recommended.

## Data Types

### CWE – Coded with Exceptions

The CWE data type specifies a coded element and its associated detail. The presence of two or more sets of equivalent codes in this data type is semantically different from a repetition of a CWE-type field. When multiple identifiers are populated with a single occurrence of the data type, the identifiers should have the same meaning, that is, they should be exact synonyms. With repetition, several distinct codes (with distinct meanings) may be transmitted. The Original Text component applies to the CWE as a whole.

Best Practices for Constructing CWE Data Type Flavors:

* When an Identifier is present, the Name of the Coding System (typical) or Coding System OID (less typical) is required to indicate the source of the code.
* When an Identifier is present, the Text is required to facilitate human understanding and troubleshooting. The curators of this data type library can’t think of a scenario where an identifier is available but a text is unavailable.
* Flavors should be constructed such that it is not possible for the Alternate equivalent codes (Alternate Identifier (CWE.4) or Second Alternate Identifier (CWE.10) to be populated when the Identifier (CWE.1) is not populated.

#### CWE\_01 - Coded with Exceptions

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Identifier (CWE.1, CWE.2 and CWE.3) is required. Support for sending an Alternate Identifier (CWE.4, CWE.5 and CWE.6) and the Original Text (CWE.9) is required but allowed to be empty.

Use of this flavor is recommended when two or more overlapping Value Sets are possible. For example, a set of national (e.g. LOINC) and local (e.g. organization specific) codes are routinely used.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Identifier | ST | R |
| 2 | Text | ST | R |
| 3 | Name of Coding System | ID | R |
| 4 | Alternate Identifier | ST | RE |
| 5 | Alternate Text | ST | C(R/X) |
| 6 | Name of Alternate Coding System | ID | C(R/X) |
| 7 | Coding System Version ID | ST | O |
| 8 | Alternate Coding System Version ID | ST | O |
| 9 | Original Text | ST | RE |
| 10 | Second Alternate Identifier | ST | O |
| 11 | Second Alternate Text | ST | O |
| 12 | Name of Second Alternate Coding System | ID | O |
| 13 | Second Alternate Coding System Version ID | ST | O |
| 14 | Coding System OID | ST | O |
| 15 | Value Set OID | ST | O |
| 16 | Value Set Version ID | DTM | O |
| 17 | Alternate Coding System OID | ST | O |
| 18 | Alternate Value Set OID | ST | O |
| 19 | Alternate Value Set Version ID | DTM | O |
| 20 | Second Alternate Coding System OID | ST | O |
| 21 | Second Alternate Value Set OID | ST | O |
| 22 | Second Alternate Value Set Version ID | DTM | O |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| CWE.5 | C(R/X) | If CWE.4 (Alternate Identifier) is valued |
| CWE.6 | C(R/X) | If CWE.4 (Alternate Identifier) is valued |

#### CWE\_02 - Coded with Exceptions

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Identifier (CWE.1, CWE.2 and CWE.3) is required. Support for sending an Alternate Identifier (CWE.4, CWE.5 and CWE.6) is optional. Support for sending the Original Text (CWE.9) is required but allowed to be empty.

Use of this flavor is recommended when only a single Value Sets is typically. In this scenario, the use of this data type lowers the burden on implementer and reduces the possibility of messaging incompatible data.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Identifier | ST | R |
| 2 | Text | ST | R |
| 3 | Name of Coding System | ID | R |
| 4 | Alternate Identifier | ST | O |
| 5 | Alternate Text | ST | O |
| 6 | Name of Alternate Coding System | ID | O |
| 7 | Coding System Version ID | ST | O |
| 8 | Alternate Coding System Version ID | ST | O |
| 9 | Original Text | ST | RE |
| 10 | Second Alternate Identifier | ST | O |
| 11 | Second Alternate Text | ST | O |
| 12 | Name of Second Alternate Coding System | ID | O |
| 13 | Second Alternate Coding System Version ID | ST | O |
| 14 | Coding System OID | ST | O |
| 15 | Value Set OID | ST | O |
| 16 | Value Set Version ID | DTM | O |
| 17 | Alternate Coding System OID | ST | O |
| 18 | Alternate Value Set OID | ST | O |
| 19 | Alternate Value Set Version ID | DTM | O |
| 20 | Second Alternate Coding System OID | ST | O |
| 21 | Second Alternate Value Set OID | ST | O |
| 22 | Second Alternate Value Set Version ID | DTM | O |

.

#### CWE\_03 - Coded with Exceptions

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Identifier (CWE.1, CWE.2 and CWE.3) and the Alternate Identifier (CWE.4, CWE.5 and CWE.6) is required but allowed to be empty. Support for sending the Original Text (CWE.9) is required when a code is not available.

Use of this flavor is recommended when two or more overlapping Value Sets are possible but when a code may not always be possible.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Identifier | ST | RE |
| 2 | Text | ST | C(R/X) |
| 3 | Name of Coding System | ID | C(R/X) |
| 4 | Alternate Identifier | ST | C(RE/X) |
| 5 | Alternate Text | ST | C(R/X) |
| 6 | Name of Alternate Coding System | ID | C(R/X) |
| 7 | Coding System Version ID | ST | O |
| 8 | Alternate Coding System Version ID | ST | O |
| 9 | Original Text | ST | C(R/RE) |
| 10 | Second Alternate Identifier | ST | O |
| 11 | Second Alternate Text | ST | O |
| 12 | Name of Second Alternate Coding System | ID | O |
| 13 | Second Alternate Coding System Version ID | ST | O |
| 14 | Coding System OID | ST | O |
| 15 | Value Set OID | ST | O |
| 16 | Value Set Version ID | DTM | O |
| 17 | Alternate Coding System OID | ST | O |
| 18 | Alternate Value Set OID | ST | O |
| 19 | Alternate Value Set Version ID | DTM | O |
| 20 | Second Alternate Coding System OID | ST | O |
| 21 | Second Alternate Value Set OID | ST | O |
| 22 | Second Alternate Value Set Version ID | DTM | O |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| CWE.2 | C(R/X) | If CWE.1 (Identifier) is valued |
| CWE.3 | C(R/X) | If CWE.1 (Identifier) is valued |
| CWE.4 | C(RE/X) | If CWE.1 (Identifier) is valued |
| CWE.5 | C(R/X) | If CWE.4 (Alternate Identifier) is valued |
| CWE.6 | C(R/X) | If CWE.4 (Alternate Identifier) is valued |
| CWE.9 | C(R/RE) | If CWE.1 (Identifier) is not valued |

#### CWE\_04 - Coded with Exceptions

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Identifier (CWE.1, CWE.2 and CWE.3) is required but allowed to be empty. Support for sending an Alternate Identifier (CWE.4, CWE.5 and CWE.6) is optional. Support for sending the Original Text (CWE.9) is required when a code is not available.

Use of this flavor is recommended when only a single Value Sets is typically. In this scenario, the use of this data type lowers the burden on implementer and reduces the possibility of messaging incompatible data.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Identifier | ST | RE |
| 2 | Text | ST | C(R/X) |
| 3 | Name of Coding System | ID | C(R/X) |
| 4 | Alternate Identifier | ST | O |
| 5 | Alternate Text | ST | O |
| 6 | Name of Alternate Coding System | ID | O |
| 7 | Coding System Version ID | ST | O |
| 8 | Alternate Coding System Version ID | ST | O |
| 9 | Original Text | ST | C(R/RE) |
| 10 | Second Alternate Identifier | ST | O |
| 11 | Second Alternate Text | ST | O |
| 12 | Name of Second Alternate Coding System | ID | O |
| 13 | Second Alternate Coding System Version ID | ST | O |
| 14 | Coding System OID | ST | O |
| 15 | Value Set OID | ST | O |
| 16 | Value Set Version ID | DTM | O |
| 17 | Alternate Coding System OID | ST | O |
| 18 | Alternate Value Set OID | ST | O |
| 19 | Alternate Value Set Version ID | DTM | O |
| 20 | Second Alternate Coding System OID | ST | O |
| 21 | Second Alternate Value Set OID | ST | O |
| 22 | Second Alternate Value Set Version ID | DTM | O |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| CWE.2 | C(R/X) | If CWE.1 (Identifier) is valued |
| CWE.3 | C(R/X) | If CWE.1 (Identifier) is valued |
| CWE.9 | C(R/RE) | If CWE.1 (Identifier) is not valued |

#### CWE\_05 - Coded with Exceptions

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Specifies a coded element and its associated detail. Support for the Identifier (CWE.1, CWE.2 and CWE.3/CWE.14) is required but allowed to be empty. Support for sending an Alternate Identifier (CWE.4, CWE.5 and CWE.6) is optional. Support for sending the Original Text (CWE.9) is required when a code is not available

<QUESTION> the current definition doesn't allow for both CWE.3 and CWE.14 to both be populated (when CWE.1 is valued) Is this a problem? Is there a way to do this? Would it be better if both were RE and we had a conformance statement saying that if CWE.1 is valued at least one of CWE.3 or CWE.14 SHALL be valued? Can we even do "at least one of" in IGAMT or is "OR" sufficient?

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Identifier | ST | RE |
| 2 | Text | ST | C(R/X) |
| 3 | Name of Coding System | ID | C(R/X) |
| 4 | Alternate Identifier | ST | O |
| 5 | Alternate Text | ST | O |
| 6 | Name of Alternate Coding System | ID | O |
| 7 | Coding System Version ID | ST | O |
| 8 | Alternate Coding System Version ID | ST | O |
| 9 | Original Text | ST | C(R/RE) |
| 10 | Second Alternate Identifier | ST | O |
| 11 | Second Alternate Text | ST | O |
| 12 | Name of Second Alternate Coding System | ID | O |
| 13 | Second Alternate Coding System Version ID | ST | O |
| 14 | Coding System OID | ST | C(R/X) |
| 15 | Value Set OID | ST | O |
| 16 | Value Set Version ID | DTM | O |
| 17 | Alternate Coding System OID | ST | O |
| 18 | Alternate Value Set OID | ST | O |
| 19 | Alternate Value Set Version ID | DTM | O |
| 20 | Second Alternate Coding System OID | ST | O |
| 21 | Second Alternate Value Set OID | ST | O |
| 22 | Second Alternate Value Set Version ID | DTM | O |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| CWE.2 | C(R/X) | If CWE.1 (Identifier) is valued |
| CWE.9 | C(R/RE) | If CWE.1 (Identifier) is not valued |
| CWE.14 | C(R/X) | [If CWE.1 (Identifier) is valued] AND [If CWE.3 (Name of Coding System) is not valued] |
| CWE.3 | C(R/X) | [If CWE.1 (Identifier) is valued] AND [If CWE.14 (Coding System OID) is not valued] |

#### CWE\_06 - Coded with Exceptions

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Identifier (CWE.1, CWE.2 and CWE.3) is required. Support for sending an Alternate Identifier (CWE.4, CWE.5 and CWE.6) is optional.

Use of this flavor is recommended when only a single Value Sets is typically. In this scenario, the use of this data type lowers the burden on implementer and reduces the possibility of messaging incompatible data.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Identifier | ST | R |
| 2 | Text | ST | R |
| 3 | Name of Coding System | ID | R |
| 4 | Alternate Identifier | ST | O |
| 5 | Alternate Text | ST | O |
| 6 | Name of Alternate Coding System | ID | O |
| 7 | Coding System Version ID | ST | O |
| 8 | Alternate Coding System Version ID | ST | O |
| 9 | Original Text | ST | O |
| 10 | Second Alternate Identifier | ST | O |
| 11 | Second Alternate Text | ST | O |
| 12 | Name of Second Alternate Coding System | ID | O |
| 13 | Second Alternate Coding System Version ID | ST | O |
| 14 | Coding System OID | ST | O |
| 15 | Value Set OID | ST | O |
| 16 | Value Set Version ID | DTM | O |
| 17 | Alternate Coding System OID | ST | O |
| 18 | Alternate Value Set OID | ST | O |
| 19 | Alternate Value Set Version ID | DTM | O |
| 20 | Second Alternate Coding System OID | ST | O |
| 21 | Second Alternate Value Set OID | ST | O |
| 22 | Second Alternate Value Set Version ID | DTM | O |

#### CWE\_07 - Coded with Exceptions

Versions: 2.5, 2.5.1, 2.6

Support for the Identifier (CWE.1, CWE.2 and CWE.3) is required. Support for sending an Alternate Identifier (CWE.4, CWE.5 and CWE.6) is optional.

Use of this flavor is recommended when only a single Value Sets is typically. In this scenario, the use of this data type lowers the burden on implementer and reduces the possibility of messaging incompatible data.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Identifier | ST | R |
| 2 | Text | ST | R |
| 3 | Name of Coding System | ID | R |
| 4 | Alternate Identifier | ST | O |
| 5 | Alternate Text | ST | O |
| 6 | Name of Alternate Coding System | ID | O |
| 7 | Coding System Version ID | ST | O |
| 8 | Alternate Coding System Version ID | ST | O |
| 9 | Original Text | ST | O |

#### CWE\_08 - Coded with Exceptions

Versions: 2.5, 2.5.1, 2.6

Support for the Identifier (CWE.1, CWE.2 and CWE.3) is required but allowed to be empty. Support for sending an Alternate Identifier (CWE.4, CWE.5 and CWE.6) is optional. Support for sending the Original Text (CWE.9) is required when a code is not available.

Use of this flavor is recommended when only a single Value Sets is typically. In this scenario, the use of this data type lowers the burden on implementer and reduces the possibility of messaging incompatible data.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Identifier | ST | RE |
| 2 | Text | ST | C(R/X) |
| 3 | Name of Coding System | ID | C(R/X) |
| 4 | Alternate Identifier | ST | O |
| 5 | Alternate Text | ST | O |
| 6 | Name of Alternate Coding System | ID | O |
| 7 | Coding System Version ID | ST | O |
| 8 | Alternate Coding System Version ID | ST | O |
| 9 | Original Text | ST | C(R/RE) |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| CWE.2 | C(R/X) | If CWE.1 (Identifier) is valued |
| CWE.3 | C(R/X) | If CWE.1 (Identifier) is valued |
| CWE.9 | C(R/RE) | If CWE.1 (Identifier) is not valued |

### CX - Extended Composite ID with Check Digit

The CX data type is used for specifying an identifier with its associated administrative details including the Assigning Authority and Identifier Type Code. The most common usage for this data type is to exchange the patient identifier in PID-3.

Best Practices for Constructing CX Data Type Flavors:

* Require both the Assigning Authority and Identifier Type Code to unambiguously identify the pool of IDs providing the identifier. The Identifier Type Code component is not a part of the actual identifier, but rather, it is metadata about the identifier. The Identifier and Assigning Authority component, together, constitute the actual identifier. The Assigning Authority represents the identifier’s name space, e.g., Healthy Hospital Medical Record Numbers, or Healthy Hospital Order Numbers. Consequently, the Identifier Type Code is technically not necessary. However, due to various naming practices, organizational mergers, and other challenges, it is not always clear through the Assigning Authority what identifier type is being indicated by the identifier name space (note that it is highly recommended that this detail be associated with the OID in the registry metadata about the OID). Therefore, to maintain forward compatibility with V3, while recognizing the current practical challenges with understanding the identifier type/namespace at hand, the recommendation is to keep the Identifier Type Code component as required.

#### CX\_01 - Extended Composite ID with Check Digit

Versions: 2.5, 2.5.1

Support for both the Assigning Authority (CX.4) and Identifier Type Code (CX.5) is required.

Use of this flavor is recommended when the ID must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by requiring the Assigning Authority (CX.4) and constraining it to require support for ISO complaint OIDs.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | ID Number | ST | R |
| 2 | Check Digit | ST | O |
| 3 | Check Digit Scheme | ID | O |
| 4 | Assigning Authority | HD\_01 | R |
| 5 | Identifier Type Code | ID | R |
| 6 | Assigning Facility | HD | O |
| 7 | Effective Date | DT | O |
| 8 | Expiration Date | DT | O |
| 9 | Assigning Jurisdiction | CWE | O |
| 10 | Assigning Agency or Department | CWE | O |

#### CX\_02 - Extended Composite ID with Check Digit

Versions: 2.5, 2.5.1

Support for the Identifier Type Code (CX.5) is required. Support for the Assigning Authority (CX.4) is required but may be empty.

Use of this flavor is recommended when the ID in need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | ID Number | ST | R |
| 2 | Check Digit | ST | O |
| 3 | Check Digit Scheme | ID | O |
| 4 | Assigning Authority | HD\_02 | RE |
| 5 | Identifier Type Code | ID | R |
| 6 | Assigning Facility | HD | O |
| 7 | Effective Date | DT | O |
| 8 | Expiration Date | DT | O |
| 9 | Assigning Jurisdiction | CWE | O |
| 10 | Assigning Agency or Department | CWE | O |

#### CX\_03 - Extended Composite ID with Check Digit

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for both the Assigning Authority (CX.4) and Identifier Type Code (CX.5) are required.

Use of this flavor is recommended when the ID in need not be globally unique, however the Assigning Authority (CX.4) must support the sending of ISO compliant OIDs as part of the HD data type.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | ID Number | ST | R |
| 2 | Identifier Check Digit | ST | O |
| 3 | Check Digit Scheme | ID | O |
| 4 | Assigning Authority | HD\_06 | R |
| 5 | Identifier Type Code | ID | R |
| 6 | Assigning Facility | HD | O |
| 7 | Effective Date | DT | O |
| 8 | Expiration Date | DT | O |
| 9 | Assigning Jurisdiction | CWE | O |
| 10 | Assigning Agency or Department | CWE | O |
| 11 | Security Check | ST | O |
| 12 | Security Check Scheme | ID | O |

#### CX\_04 - Extended Composite ID with Check Digit

Versions: 2.6

Support for the identifier Type Code (CX.5) is required. Support for the Assigning Authority (CX.4) is required but may be empty.

Use of this flavor is recommended when the ID must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by constraining the Assigning Authority (CX.4) to require support for ISO complaint OIDs.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | ID Number | ST | R |
| 2 | Identifier Check Digit | ST | O |
| 3 | Check Digit Scheme | ID | O |
| 4 | Assigning Authority | HD\_05 | RE |
| 5 | Identifier Type Code | ID | R |
| 6 | Assigning Facility | HD | O |
| 7 | Effective Date | DT | O |
| 8 | Expiration Date | DT | O |
| 9 | Assigning Jurisdiction | CWE | O |
| 10 | Assigning Agency or Department | CWE | O |

### DTM – Date/time

The DTM data type is used for specifying a point in time using a 24-hour clock notation.

Best Practices for Constructing DTM Data Type Flavors:

* It is strongly recommended that the time zone offset always be included in the DTM if the value includes hours, minutes, seconds, etc.

Best Practices for Implementing DTM Data Type Flavors:

* Remember that when an element in the DTM data has a Usage of RE, the system receiving the value must be able to support the receipt of data both with and without the element. This can place additional burden on the receiving system. For example, when the year has a Usage of R but the month and day have a Usage of RE, receiving systems must be capable of supporting just a year, a year and a month and a year, a month and a day.

#### DTM\_01 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year is required. Support for the month and day is required but may be empty.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | RE |  |
| 3 | DD | C(RE/X) | If MM(Month) is valued. |
| 4 | HH | C(O/X) | If DD(Day) is valued. |
| 5 | MM | C(O/X) | If HH(Hour) is valued. |
| 6 | SS | C(O/X) | If MM(Minute) is valued. |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | O |  |

#### DTM\_02 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year is required. Support for the month, day, hour and minutes is required but may be empty.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | RE |  |
| 3 | DD | C(RE/X) | If MM(Month) is valued. |
| 4 | HH | C(RE/X) | If DD(Day) is valued. |
| 5 | MM | C(RE/X) | If HH(Hour) is valued. |
| 6 | SS | C(O/X) | If MM(Minute) is valued. |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | O |  |

#### DTM\_03 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year is required. Support for the month, day, hour and minutes is required but may be empty. Support for the time zone offset is required when the hour element is valued.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | RE |  |
| 3 | DD | C(RE/X) | If MM(Month) is valued. |
| 4 | HH | C(RE/X) | If DD(Day) is valued. |
| 5 | MM | C(RE/X) | If HH(Hour) is valued. |
| 6 | SS | C(O/X) | If MM(Minute) is valued. |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | C(R/X) | If HH(Hour) is valued. |

#### DTM\_05 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year, month and day is required.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | R |  |
| 3 | DD | R |  |
| 4 | HH | O |  |
| 5 | MM | C(O/X) | If HH(Hour) is valued. |
| 6 | SS | C(O/X) | If MM(Minute) is valued. |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | O |  |

#### DTM\_06 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year, month and day is required. Support for the hour and minutes is required but may be empty.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | R |  |
| 3 | DD | R |  |
| 4 | HH | RE |  |
| 5 | MM | C(RE/X) | If HH(Hour) is valued. |
| 6 | SS | C(O/X) | If MM(Minute) is valued. |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | O |  |

#### DTM\_07 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year, month and day is required. Support for the hour and minutes is required but may be empty. Support for the time zone offset is required when the hour element is valued.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | R |  |
| 3 | DD | R |  |
| 4 | HH | RE |  |
| 5 | MM | C(RE/X) | If HH(Hour) is valued. |
| 6 | SS | C(O/X) | If MM(Minute) is valued. |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | C(R/X) | If HH(Hour) is valued. |

#### DTM\_08 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year, month, day, hour and minutes is required.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | R |  |
| 3 | DD | R |  |
| 4 | HH | R |  |
| 5 | MM | R |  |
| 6 | SS | C(O/X) | If MM(Minute) is valued. |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | O |  |

#### DTM\_09 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year, month, day, hour, minutes and time zone offset is required.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | R |  |
| 3 | DD | R |  |
| 4 | HH | R |  |
| 5 | MM | R |  |
| 6 | SS | C(O/X) | If MM(Minute) is valued. |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | R |  |

#### DTM\_10 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year, month, day, hour, minutes and seconds is required.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | R |  |
| 3 | DD | R |  |
| 4 | HH | R |  |
| 5 | MM | R |  |
| 6 | SS | R |  |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | O |  |

#### DTM\_11 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year, month, day, hour, minutes, seconds and time zone offset is required.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | R |  |
| 3 | DD | R |  |
| 4 | HH | R |  |
| 5 | MM | R |  |
| 6 | SS | R |  |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | R |  |

#### DTM\_12 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year is required including a value of '0000' to indicate a date/time is not available. Support for the month and day when a date/time is available. Support for hour and minutes is required but may be empty when a date/time is available.

Use of this flavor is recommended only when the base standard requires the field to be populated but the use case may not always allow a date/time to be known.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | C(R/X) | If Year(Year) is not literal value '0000'. |
| 3 | DD | C(R/X) | If Year(Year) is not literal value '0000'. |
| 4 | HH | C(RE/X) | If Year(Year) is not literal value '0000'. |
| 5 | MM | C(RE/X) | If Year(Year) is not literal value '0000'. |
| 6 | SS | C(O/X) | If Year(Year) is not literal value '0000'. |
| 7 | s | C(O/X) | If Year(Year) is not literal value '0000'. |
| 8 | s | C(O/X) | If Year(Year) is not literal value '0000'. |
| 9 | s | C(O/X) | If Year(Year) is not literal value '0000'. |
| 10 | s | C(O/X) | If Year(Year) is not literal value '0000'. |
| 11 | ZZZZ | O |  |

#### DTM\_13 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year is required including a value of '0000' to indicate a date/time is not available. Support for the month and day when a date/time is available. Support for hour and minutes is required but may be empty when a date/time is available. Support for the time zone offset is required when the hour element is valued.

Use of this flavor is recommended only when the base standard requires the field to be populated but the use case may not always allow a date/time to be known.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | C(R/X) | If Year(Year) is not literal value '0000'. |
| 3 | DD | C(R/X) | If Year(Year) is not literal value '0000'. |
| 4 | HH | C(RE/X) | If Year(Year) is not literal value '0000'. |
| 5 | MM | C(RE/X) | If Year(Year) is not literal value '0000'. |
| 6 | SS | C(O/X) | If Year(Year) is not literal value '0000'. |
| 7 | s | C(O/X) | If Year(Year) is not literal value '0000'. |
| 8 | s | C(O/X) | If Year(Year) is not literal value '0000'. |
| 9 | s | C(O/X) | If Year(Year) is not literal value '0000'. |
| 10 | s | C(O/X) | If Year(Year) is not literal value '0000'. |
| 11 | ZZZZ | C(R/X) | If HH(Hour) is valued. |

#### DTM\_14 - Date/Time

Version: 2.5, 2.5.1, 2.6, 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the year and month is required.

**Data Type Definition**

| **#** | **Value** | **Usage** | **Predicate** |
| --- | --- | --- | --- |
| 1 | YYYY | R |  |
| 2 | MM | R |  |
| 3 | DD | O |  |
| 4 | HH | C(O/X) | If DD(Day) is valued. |
| 5 | MM | C(O/X) | If HH(Hour) is valued. |
| 6 | SS | C(O/X) | If MM(Minute) is valued. |
| 7 | s | C(O/X) | If SS(Second) is valued. |
| 8 | s | C(O/X) | If s(1/10 second) is valued. |
| 9 | s | C(O/X) | If s(1/100 second) is valued. |
| 10 | s | C(O/X) | If s(1/1000 second) is valued. |
| 11 | ZZZZ | O |  |

### ED - Encapsulated Data

The ED data type is used for transmitting encapsulated data such as a pre-formatted version of a report (e.g., a PDF file).

#### ED\_01 - Encapsulated Data

Versions: 2.5, 2.5.1

Support for the Type of Date (ED.1), Encoding (ED.4) and Data (ED.5) is required. Support for the Data Subtype (ED.3) is required but may be empty.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Source Application | HD | O |
| 2 | Type of Data | ID | R |
| 3 | Data Subtype | ID | RE |
| 4 | Encoding | ID | R |
| 5 | Data | TX | R |

### EI - Entity Identifier

The EI data type is used for specifying a given entity within a specified series of identifiers. Components 2 through 4, are known as the assigning authority and identify the machine/system responsible for generating the identifier in component 1, functioning similarly to the hierarchic designator (HD) data type.

Best Practices for Constructing EI Data Type Flavors:

* When constraining the Universal ID (EI.3) to a limited set of acceptable values (for example, requiring it to be an ISO compliant OID), the Universal ID Type (EI.4) should be constrained in the data type flavor using a Conformance Statement.

#### EI\_01 - Entity Identifier

Versions: 2.5, 2.5.1, 2.6

Support for both the Universal ID (EI.3) and Universal ID Type (EI.4) is required. Support for the Namespace ID (EI.2) is required but may be empty.

Use of this flavor is recommended when the ID must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by requiring the Universal ID (EI.3) and constraining it to be an ISO complaint OID.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Entity Identifier | ST | R |
| 2 | Namespace ID | IS | RE |
| 3 | Universal ID | ST | R |
| 4 | Universal ID Type | ID | R |

**Conformance Statements**

| **ID** | **Description** |
| --- | --- |
| EI\_01-01 | The value of EI-3 (Universal ID) SHALL be formatted with 'ISO-compliant OID'. |
| EI\_01-02 | The value of EI-4 (Universal ID Type) SHALL be 'ISO'. |

#### EI\_02 - Entity Identifier

Versions: 2.5, 2.5.1, 2.6

Support for both the Namespace ID (EI.2) and the Universal ID (EI.3)/Universal ID Type (EI.4) is required, but in a given instance of the data type flavor, only one of the two must be present.

Use of this flavor is recommended when the ID in need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Entity Identifier | ST | R |
| 2 | Namespace ID | IS | C(R/O) |
| 3 | Universal ID | ST | C(R/O) |
| 4 | Universal ID Type | ID | C(R/X) |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| EI.2 | C(R/O) | If EI.3 (Universal ID) is not valued |
| EI.3 | C(R/O) | If EI.2 (Namespace ID) is not valued |
| EI.4 | C(R/X) | If EI.3 (Universal ID) is valued |

#### EI\_03 - Entity Identifier

Versions: 2.5, 2.5.1, 2.6

Support for both the Universal ID (EI.3) and Universal ID Type (EI.4) is required. Support for the Namespace ID (EI.2) is required but may be empty.

Use of this flavor is recommended when the ID in need not be globally unique but a Universal ID is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Entity Identifier | ST | R |
| 2 | Namespace ID | IS | RE |
| 3 | Universal ID | ST | R |
| 4 | Universal ID Type | ID | R |

#### EI\_04 - Entity Identifier

Versions: 2.5, 2.5.1, 2.6

Support for Namespace ID (EI.2) is required. Support for the Universal ID (EI.3)/Universal ID Type (EI.4) is required but allowed to be empty, however, when the Universal ID is populated, it is constrained to an ISO compliant OID.

Use of this flavor is recommended when the ID in need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Entity Identifier | ST | R |
| 2 | Namespace ID | IS | R |
| 3 | Universal ID | ST | RE |
| 4 | Universal ID Type | ID | C(R/X) |

**Conformance Statements**

| **ID** | **Description** |
| --- | --- |
| EI\_04-01 | The value of EI-3 (Universal ID) SHALL be formatted with 'ISO-compliant OID'. |
| EI\_04-02 | The value of EI-4 (Universal ID Type) SHALL be 'ISO'. |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| EI.4 | C(R/X) | If EI.3 (Universal ID) is valued |

#### EI\_05 - Entity Identifier

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Namespace ID (EI.2) is required.

Use of this flavor is recommended when the ID in need not be globally unique and Universal IDs are not expected to be used.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Entity Identifier | ST | R |
| 2 | Namespace ID | IS | R |
| 3 | Universal ID | ST | O |
| 4 | Universal ID Type | ID | O |

#### EI\_06 - Entity Identifier

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Namespace ID (EI.2) is required. Support for both the Universal ID (EI.3) and Universal ID Type (EI.4) is required but may be empty. When the Universal ID is populated, it is constrained to an ISO compliant OID.

Use of this flavor is recommended when the ID in need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Entity Identifier | ST | R |
| 2 | Namespace ID | IS | R |
| 3 | Universal ID | ST | RE |
| 4 | Universal ID Type | ID | C(R/X) |

**Conformance Statements**

| **ID** | **Description** |
| --- | --- |
| EI\_05-01 | The value of EI-3 (Universal ID) SHALL be formatted with 'ISO-compliant OID'. |
| EI\_05-02 | The value of EI-4 (Universal ID Type) SHALL be 'ISO'. |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| EI.4 | C(R/X) | If EI.3 (Universal ID) is valued |

### EIP - Entity Identifier Pair

The EIP data type is used for specifying an identifier assigned to an entity by either the placer or the filler system. If both components are populated the identifiers must refer to the same entity.

#### EIP\_01 - Entity Identifier Pair

Versions: 2.5, 2.5.1, 2.6

Support for both the Placer (EIP.1) and the Filler (EIP.2) Assigned Identifiers is required but allowed to be empty, but in a given instance of the data type flavor, only one of the two must be present.

Use of this flavor is recommended when the ID must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by the use of an EI data type flavor requiring the use of ISO compliant OIDs.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Placer Assigned Identifier | EI\_01 | RE |
| 2 | Filler Assigned Identifier | EI\_01 | C(R/RE) |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| EIP.2 | C(R/RE) | If EIP.1 (Placer Assigned Identifier) is not valued |

#### EIP\_02 - Entity Identifier Pair

Versions: 2.5, 2.5.1, 2.6

Support for both the Placer (EIP.1) and the Filler (EIP.2) Assigned Identifiers is required but allowed to be empty, but in a given instance of the data type flavor, only one of the two must be present.

Use of this flavor is recommended when the ID in need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Placer Assigned Identifier | EI\_02 | RE |
| 2 | Filler Assigned Identifier | EI\_02 | C(R/RE) |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| EIP.2 | C(R/RE) | If EIP.1 (Placer Assigned Identifier) is not valued |

#### EIP\_03 - Entity Identifier Pair

Versions: 2.5, 2.5.1, 2.6

Support for both the Placer (EIP.1) and the Filler (EIP.2) Assigned Identifiers is required.

Use of this flavor is recommended when the ID must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by the use of an EI data type flavor requiring the use of ISO compliant OIDs.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Placer Assigned Identifier | EI\_01 | R |
| 2 | Filler Assigned Identifier | EI\_01 | R |

#### EIP\_04 - Entity Identifier Pair

Versions: 2.5, 2.5.1, 2.6

Support for both the Placer (EIP.1) and the Filler (EIP.2) Assigned Identifiers is required.

Use of this flavor is recommended when the ID in need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Placer Assigned Identifier | EI\_02 | R |
| 2 | Filler Assigned Identifier | EI\_02 | R |

#### EIP\_05 - Entity Identifier Pair

Versions: 2.5, 2.5.1, 2.6

Support for both the Placer (EIP.1) and the Filler (EIP.2) Assigned Identifiers is required.

Use of this flavor is recommended when the ID in need not be globally unique but a Universal ID is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Placer Assigned Identifier | EI\_03 | R |
| 2 | Filler Assigned Identifier | EI\_03 | R |

### ERL - Error Location

The ERL data type is used for specifying the segment and its constituent where an error has occurred

Best Practices for Constructing ERL Data Type Flavors:

* The curators of this data type library can’t think of a scenario where the technical format of reporting errors (using the ERL data type) would be influenced by the use case(s) in the implementation guide. As such, alternative ERL data type flavors should not be required.

#### ERL\_01 - Error Location

Versions: 2.5, 2.5.1, 2.6

Support for the Segment ID (ERL.1) and Segment Sequence (ERL.2) is required. Support for all other components is required but may be empty.

Use of this flavor is recommended in all scenarios.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Segment ID | ST | R |
| 2 | Segment Sequence | NM | R |
| 3 | Field Position | NM | C(R/RE) |
| 4 | Field Repetition | NM | RE |
| 5 | Component Number | NM | C(R/RE) |
| 6 | Sub-Component Number | NM | RE |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| ERL.3 | C(R/RE) | If ERL.5 (Component Number) is valued |
| ERL.5 | C(R/RE) | If ERL.6 (Sub-Component Number) is valued |

**Component Definitions**   
  
**ERL.2 : Segment Sequence**

Absolute position of this segment in the message (e.g. 3rd NTE in message, regardless of the number or type of intervening segments).

**ERL.3: Field Position**   
ERL.3 should not be populated if the error refers to the whole segment.

**ERL.5 : Field Repetition**

If not specified, occurrence is assumed to be 1.

**ERL.5: Component Number**   
ERL.5 should not be populated if the error refers to the whole field.

**ERL.6: Sub-Component Number**   
ERL.6 should not be populated if the error refers to the whole component.

#### ERL\_02 - Error Location

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Segment ID (ERL.1) and Segment Sequence (ERL.2) is required. Support for all other components is required but may be empty.

Use of this flavor is recommended in all scenarios.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Segment ID | ST | R |
| 2 | Segment Sequence | NM | R |
| 3 | Field Position | NM | C(R/RE) |
| 4 | Field Repetition | NM | RE |
| 5 | Component Number | NM | C(R/RE) |
| 6 | Sub-Component Number | NM | RE |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| ERL.3 | C(R/RE) | If ERL.5 (Component Number) is valued |
| ERL.5 | C(R/RE) | If ERL.6 (Sub-Component Number) is valued |

**Component Definitions**   
  
**ERL.2 : Segment Sequence**

Absolute position of this segment in the message (e.g. 3rd NTE in message, regardless of the number or type of intervening segments).

**ERL.3 : Field Position**   
ERL.3 should not be populated if the error refers to the whole segment.

**ERL.5 : Field Repetition**

If not specified, occurrence is assumed to be 1.

**ERL.5 : Component Number**   
ERL.5 should not be populated if the error refers to the whole field.

**ERL.6 : Sub-Component Number**   
ERL.6 should not be populated if the error refers to the whole component.

### FN - Family Name

The FN data type is used for specifying the surname of a person.

#### FN\_01 - Family Name

Versions: 2.5, 2.5.1, 2.6

Support for the Surname (FN.1) is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Surname | ST | R |
| 2 | Own Surname Prefix | ST | O |
| 3 | Own Surname | ST | O |
| 4 | Surname Prefix from Partner/Spouse | ST | O |
| 5 | Surname from Partner/Spouse | ST | O |

#### FN\_02 - Family Name

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Surname (FN.1) is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Surname | ST | R |
| 2 | Own Surname Prefix | ST | O |
| 3 | Own Surname | ST | O |
| 4 | Surname Prefix from Partner/Spouse | ST | O |
| 5 | Surname from Partner/Spouse | ST | O |

### HD - Hierarchic Designator

The HD data type is used for specifying an entity that has responsibility for managing or assigning a defined set of instance identifiers. The HD is a group of two identifiers: a local identifier defined by the first component and a universal identifier defined by the second and third components

Best Practices for Constructing HD Data Type Flavors:

* Per the base standard, the second and third components must either both be valued (both non-null), or both be not valued (both null).

Best Practices for Implementing DTM Data Type Flavors:

* If all three components of the HD are valued, the entity identified by the first component is the same as the entity identified by components two and three taken together.

#### HD\_01 - Hierarchic Designator

Versions: 2.5, 2.5.1

Support for both the Universal ID (HD.2) and Universal ID Type (HD.3) is required. Support for the Namespace ID (HD.1) is required but may be empty.

Use of this flavor is recommended when the entity ID must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by requiring the Universal ID (HD.2) and constraining it to be an ISO complaint OID.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Namespace ID | IS | RE |
| 2 | Universal ID | ST | R |
| 3 | Universal ID Type | ID | R |

**Conformance Statements**

| **ID** | **Description** |
| --- | --- |
| HD\_01-01 | The value of HD.2 (Universal ID) SHALL be formatted with an ISO-compliant OID. |
| HD\_01-02 | The value of HD.3 (Universal ID Type) SHALL be 'ISO'. |

#### HD\_02 - Hierarchic Designator

Versions: 2.5, 2.5.1

Support for both the Namespace ID (HD.1) and the Universal ID (HD.2)/Universal ID Type (HD.3) is required, but in a given instance of the data type flavor, only one of the two must be present.

Use of this flavor is recommended when the ID in need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Namespace ID | IS | C(R/O) |
| 2 | Universal ID | ST | C(R/O) |
| 3 | Universal ID Type | ID | C(R/X) |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| HD.1 | C(R/O) | If HD.2 (Universal ID) is not valued |
| HD.2 | C(R/O) | If HD.1 (Namespace ID) is not valued |
| HD.3 | C(R/X) | If HD.2 (Universal ID) is valued |

#### HD\_03 - Hierarchic Designator

Versions: 2.5, 2.5.1

Support for both the Universal ID (HD.2) and Universal ID Type (HD.3) is required. Support for the Namespace ID (HD.1) is required but may be empty.

Use of this flavor is recommended when the ID in need not be globally unique but a Universal ID is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Namespace ID | IS | RE |
| 2 | Universal ID | ST | R |
| 3 | Universal ID Type | ID | R |

#### HD\_04 - Hierarchic Designator

Versions: 2.6

Support for Namespace ID (HD.1) is required. Support for the Universal ID (HD.2) and Universal ID Type (HD.3) is required but allowed to be empty, however, when the Universal ID is populated, it is constrained to an ISO compliant OID.

Use of this flavor is recommended when the ID in need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Namespace ID | IS | R |
| 2 | Universal ID | ST | RE |
| 3 | Universal ID Type | ID | C(R/X) |

**Conformance Statements**

| **ID** | **Description** |
| --- | --- |
| HD\_07-01 | The value of HD.2 (Universal ID) SHALL be formatted with an ISO-compliant OID. |
| HD\_07-02 | The value of HD.3 (Universal ID Type) SHALL be 'ISO'. |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| HD.3 | C(R/X) | If HD.2 (Universal ID) is valued |

#### HD\_05 - Hierarchic Designator

Versions: 2.6

Support for Universal ID (HD.2) and Universal ID Type (HD.3) is required.

Use of this flavor is recommended when the entity ID must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by requiring the Universal ID (HD.2) and constraining it to be an ISO complaint OID.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Namespace ID | IS | O |
| 2 | Universal ID | ST | R |
| 3 | Universal ID Type | ID | R |

**Conformance Statements**

| **ID** | **Description** |
| --- | --- |
| HD\_07-01 | The value of HD.2 (Universal ID) SHALL be formatted with an ISO-compliant OID. |
| HD\_07-02 | The value of HD.3 (Universal ID Type) SHALL be 'ISO'. |

#### HD\_06 - Hierarchic Designator

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Namespace ID (HD.1) is required. Support for both the Universal ID (HD.2) and Universal ID Type (HD.3) is required but may be empty. When the Universal ID is populated, it is constrained to an ISO compliant OID.

Use of this flavor is recommended when the ID in need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Namespace ID | IS | R |
| 2 | Universal ID | ST | RE |
| 3 | Universal ID Type | ID | C(R/X) |

**Conformance Statements**

| **ID** | **Description** |
| --- | --- |
| HD\_04-1 | The value of HD.2 (Universal ID) SHALL be formatted with ISO-compliant OID. |
| HD\_04-2 | The value of HD.3 (Universal ID Type) SHALL be 'ISO'. |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| HD.3 | C(R/X) | If HD.2 (Universal ID) is valued |

### MSG - Message Type

The MSG data type is used for specifying the message type, trigger event, and the message structure ID for the message

Best Practices for Constructing EI Data Type Flavors:

* The base standard requires all 3 MSG components and cannot be further constrained. As such, alternative MSG data type flavors should not be required.

#### MSG\_01 - Message Type

Versions: 2.5, 2.5.1, 2.6

Support for the all components is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Message Code | ID | R |
| 2 | Trigger Event | ID | R |
| 3 | Message Structure | ID | R |

#### MSG\_02 - Message Type

Versions: 2.8.2

Support for the all components is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Message Code | ID | R |
| 2 | Trigger Event | ID | R |
| 3 | Message Structure | ID | R |

### OG - Observation Grouper

The OG data type is used for specifying the relationship of the observation/result segments (OBX) within a message.

#### OG\_01 - Observation Grouper

Versions: 2.8.2

Support for both the Group (OG.2) and Sequence (OG.3) is required. Support for the Identifier (OG.4) is required but may be empty.

Use of this flavor is recommended when enhanced observation (OBX segment) usage is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Original Sub-Identifier | ST | O |
| 2 | Group | NM | R |
| 3 | Sequence | NM | R |
| 4 | Identifier | ST | RE |

#### OG\_02 - Observation Grouper

Versions: 2.8.2

Support for both the Original Sub-identifier (OG.1) is required.

Use of this flavor is recommended when enhanced observation (OBX segment) usage is not required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Original Sub-Identifier | ST | R |
| 2 | Group | NM | O |
| 3 | Sequence | NM | O |
| 4 | Identifier | ST | O |

### PT - Processing Type

The PT data type is used for specifying the processing type of the message

Best Practices for Constructing PT Data Type Flavors:

* The curators of this data type library can’t think of a scenario where the optional Processing Mode (PT.2) component would be influenced by the use case(s) in the implementation guide. As such, alternative PT data type flavors should not be required.

#### PT\_01 - Processing Type

Versions: 2.5, 2.5.1, 2.6

Support for the Processing ID (PT.1) is required

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Processing ID | ID | R |
| 2 | Processing Mode | ID | O |

#### PT\_02 - Processing Type

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Processing ID (PT.1) is required

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Processing ID | ID | R |
| 2 | Processing Mode | ID | O |

### SAD - Street Address

The SAD data type is used for specifying the street address in the XAD data type.

Best Practices for Constructing SAD Data Type Flavors:

* Exchange of the Street Name (SAD.2) and Dwelling Number (SAD.3) is very atypical. As such, alternative MSG data type flavors should not be required.

#### SAD\_01 - Street Address

Versions: 2.5, 2.5.1, 2.6

Support for the Street or Mailing Address (SAD.1) is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Street or Mailing Address | ST | R |
| 2 | Street Name | ST | O |
| 3 | Dwelling Number | ST | O |

#### SAD\_02 - Street Address

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Street or Mailing Address (SAD.1) is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Street or Mailing Address | ST | R |
| 2 | Street Name | ST | O |
| 3 | Dwelling Number | ST | O |

### SN - Structured Numeric

The SN data type is used to unambiguously express numeric values along with qualifications. This enables receiving systems to store the components separately, and facilitates the use of numeric database queries.

Best Practices for Constructing SN Data Type Flavors:

* Given the inter-relatedness of the 4 components of the SN data type, the curators of this data type library can’t think of a scenario where variations in requirements would be required by a use case. As such, alternative SN data type flavors should not be required

There is no known use for “.” as the Separator/Suffix (SN.3), therefore it should not be included in a data type flavor without a specific use case.

#### SN\_01 - Structured Numeric

Versions: 2.5, 2.5.1, 2.6

Support for both the Num1 (SN.2) and Separator/Suffix (SN.3) is required. Support for the Comparator (SN.1) and Num2 (SN.4) is required but may be empty.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Comparator | ST | RE |
| 2 | Num1 | NM | R |
| 3 | Separator/Suffix | ST | C(R/O) |
| 4 | Num2 | NM | RE |

**Conformance Statements**

| **ID** | **Description** |
| --- | --- |
| SN\_01-01 | The value of SN-1 (Comparator) SHALL be one of list values: >, <, >=, <=, =, <>. |
| SN\_01-02 | The value of SN-3 (Separator/Suffix) SHALL be one of list values: -, +, /, :. |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| SN.3 | C(R/O) | [If SN.2 (Num1) is valued] AND [If SN.4 (Num2) is valued] |

### VID - Version Identifier

The VID data type is used for specifying the HL7 version of the message.

Best Practices for Constructing VID Data Type Flavors:

* Exchange of the Internationalization Code (VID.2) and International Version (VID.3) is very atypical and should not be required without a specific use case.

#### VID\_01 - Version Identifier

Versions: 2.5, 2.5.1

Support for the Version ID (VID.1) is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Version ID | ID | R |
| 2 | Internationalization Code | CE | O |
| 3 | International Version ID | CE | O |

#### VID\_02 - Version Identifier

Versions: 2.6

Support for the Version ID (VID.1) is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Version ID | ID | R |
| 2 | Internationalization Code | CWE | O |
| 3 | International Version ID | CWE | O |

#### VID\_03 - Version Identifier

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for the Version ID (VID.1) is required.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Version ID | ID | R |
| 2 | Internationalization Code | CWE | O |
| 3 | International Version ID | CWE | O |

### XAD - Extended Address

The XAD data type is used for specifying the address of a person, place or organization plus associated information.

Best Practices for Constructing XAD Data Type Flavors:

* At least one component should be required, if only the Address Type (XAD.7).

Best Practices for Implementing XAD Data Type Flavors:

* Consider whether or not core address components (street address, city, etc) will always be available. This may be the case for organization but not always for persons. For example, an unconscious patient may be the subject of a message but unable to provide an address.

#### XAD\_01 - Extended Address

Versions: 2.5, 2.5.1

Support for core components is required but allowed to be empty.

Use of this flavor is recommended when specific components of the address cannot be guaranteed.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Street Address | SAD\_01 | RE |
| 2 | Other Designation | ST | RE |
| 3 | City | ST | RE |
| 4 | State or Province | ST | RE |
| 5 | Zip or Postal Code | ST | RE |
| 6 | Country | ID | RE |
| 7 | Address Type | ID | RE |
| 8 | Other Geographic Designation | ST | O |
| 9 | County/Parish Code | IS | RE |
| 10 | Census Tract | IS | O |
| 11 | Address Representation Code | ID | O |
| 12 | Address Validity Range | DR\_DTM | X |
| 13 | Effective Date | TS | O |
| 14 | Expiration Date | TS | O |

#### XAD\_02 - Extended Address

Versions: 2.5, 2.5.1

Support for Street Address (XAD.1), City (XAD.3), State/Province (XAD.4), Zip/Postal Code (XAD.5) and County/Parish Code (XAD.9) is required. Support for other core components is required but allowed to be empty.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Street Address | SAD\_01 | R |
| 2 | Other Designation | ST | RE |
| 3 | City | ST | R |
| 4 | State or Province | ST | R |
| 5 | Zip or Postal Code | ST | R |
| 6 | Country | ID | RE |
| 7 | Address Type | ID | RE |
| 8 | Other Geographic Designation | ST | O |
| 9 | County/Parish Code | IS | R |
| 10 | Census Tract | IS | O |
| 11 | Address Representation Code | ID | O |
| 12 | Address Validity Range | DR\_DTM | X |
| 13 | Effective Date | TS | O |
| 14 | Expiration Date | TS | O |

#### XAD\_03 - Extended Address

Versions: 2.5, 2.5.1

Support for Street Address (XAD.1), City (XAD.3), State/Province (XAD.4) and Zip/Postal Code (XAD.5) is required. Support for other core components is required but allowed to be empty.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Street Address | SAD\_01 | R |
| 2 | Other Designation | ST | RE |
| 3 | City | ST | R |
| 4 | State or Province | ST | R |
| 5 | Zip or Postal Code | ST | R |
| 6 | Country | ID | RE |
| 7 | Address Type | ID | RE |
| 8 | Other Geographic Designation | ST | O |
| 9 | County/Parish Code | IS | RE |
| 10 | Census Tract | IS | O |
| 11 | Address Representation Code | ID | O |
| 12 | Address Validity Range | DR\_DTM | X |
| 13 | Effective Date | TS | O |
| 14 | Expiration Date | TS | O |

#### XAD\_04 - Extended Address

Versions: 2.5, 2.5.1

Support for core components is required but allowed to be empty.

Use of this flavor is recommended when specific components of the address cannot be guaranteed and when exchanging less granular addresses such as jurisdictional data where only a state or country is likely to be known.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Street Address | SAD\_01 | O |
| 2 | Other Designation | ST | O |
| 3 | City | ST | RE |
| 4 | State or Province | ST | RE |
| 5 | Zip or Postal Code | ST | RE |
| 6 | Country | ID | RE |
| 7 | Address Type | ID | O |
| 8 | Other Geographic Designation | ST | O |
| 9 | County/Parish Code | IS | RE |
| 10 | Census Tract | IS | O |
| 11 | Address Representation Code | ID | O |
| 12 | Address Validity Range | DR\_DTM | X |
| 13 | Effective Date | TS | O |
| 14 | Expiration Date | TS | O |

#### XAD\_05 - Extended Address

Versions: 2.6

Support for Country (XAD.6) and Address Type (XAD.7) is required. Support for State/Province (XAD.4) and Other Geographical Designation (XAD.8) is required but allowed to be empty.

Use of this flavor is recommended when exchanging less granular addresses such as birth place where only a state or country is likely to be known.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Street Address | SAD\_01 | O |
| 2 | Other Designation | ST | O |
| 3 | City | ST | O |
| 4 | State or Province | ST | RE |
| 5 | Zip or Postal Code | ST | O |
| 6 | Country | ID | R |
| 7 | Address Type | ID | R |
| 8 | Other Geographic Designation | ST | RE |
| 9 | County/Parish Code | IS | O |
| 10 | Census Tract | IS | O |
| 11 | Address Representation Code | ID | O |
| 12 | Address Validity Range | DR\_DTM | X |
| 13 | Effective Date | DTM | O |
| 14 | Expiration Date | DTM | O |
| 15 | Expiration Reason | CWE | O |
| 16 | Temporary Indicator | ID | O |
| 17 | Bad Address Indicator | ID | O |
| 18 | Address Usage | ID | O |
| 19 | Addressee | ST | O |
| 20 | Comment | ST | O |
| 21 | Preference Order | NM | O |
| 22 | Protection Code | CWE | O |
| 23 | Address Identifier | EI | O |

#### XAD\_06 - Extended Address

Versions: 2.6

Support for Country (XAD.6) and Address Type (XAD.7) is required. Support for other core components is required but allowed to be empty.

Use of this flavor is recommended when exchanging less granular addresses such as birth place where only a state or country is likely to be known.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Street Address | SAD\_01 | RE |
| 2 | Other Designation | ST | RE |
| 3 | City | ST | RE |
| 4 | State or Province | ST | RE |
| 5 | Zip or Postal Code | ST | RE |
| 6 | Country | ID | R |
| 7 | Address Type | ID | R |
| 8 | Other Geographic Designation | ST | RE |
| 9 | County/Parish Code | IS | O |
| 10 | Census Tract | IS | O |
| 11 | Address Representation Code | ID | O |
| 12 | Address Validity Range | DR\_DTM | X |
| 13 | Effective Date | DTM | O |
| 14 | Expiration Date | DTM | O |
| 15 | Expiration Reason | CWE | O |
| 16 | Temporary Indicator | ID | O |
| 17 | Bad Address Indicator | ID | O |
| 18 | Address Usage | ID | O |
| 19 | Addressee | ST | O |
| 20 | Comment | ST | O |
| 21 | Preference Order | NM | O |
| 22 | Protection Code | CWE | O |
| 23 | Address Identifier | EI | O |

#### XAD\_07 - Extended Address

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for Address Type (XAD.7) is required. Support for other core components is required but allowed to be empty.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Street Address | SAD | RE |
| 2 | Other Designation | ST | RE |
| 3 | City | ST | RE |
| 4 | State or Province | ST | RE |
| 5 | Zip or Postal Code | ST | RE |
| 6 | Country | ID | RE |
| 7 | Address Type | ID | R |
| 8 | Other Geographic Designation | ST | O |
| 9 | County/Parish Code | CWE | O |
| 10 | Census Tract | CWE | O |
| 11 | Address Representation Code | ID | O |
| 12 | Address Validity Range | - | X |
| 13 | Effective Date | DTM | O |
| 14 | Expiration Date | DTM | O |
| 15 | Expiration Reason | CWE | O |
| 16 | Temporary Indicator | ID | O |
| 17 | Bad Address Indicator | ID | O |
| 18 | Address Usage | ID | O |
| 19 | Addressee | ST | O |
| 20 | Comment | ST | O |
| 21 | Preference Order | NM | O |
| 22 | Protection Code | CWE | O |
| 23 | Address Identifier | EI | O |

### XCN - Extended Composite ID Number and Name for Persons

The XCN data type is used for specifying the ID number and/or name of a person. Despite XCN.1 being named ID Number in early base versions, this component is an ST string data type, not numeric, so the component is not limited to just numbers

Best Practices for Constructing XCN Data Type Flavors:

* The data type should be constructed so as to require a person identifier or person name in a given instance of the data type.

#### XCN\_01 - Extended Composite ID Number and Name for Persons

Versions: 2.5, 2.5.1

Support for Assigning Authority (XCN.9) and Identifier Type Code (XCN.13) is required. Support for other core components is required but allowed to be empty.

Use of this flavor is recommended when the person identifier must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by requiring the Assigning Authority (XCN.9) when an ID is being exchanged and constraining it to require support for ISO complaint OIDs.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | ID Number | ST | RE |
| 2 | Family Name | FN\_01 | RE |
| 3 | Given Name | ST | RE |
| 4 | Second and Further Given Names or Initials Thereof | ST | RE |
| 5 | Suffix (e.g., JR or III) | ST | RE |
| 6 | Prefix (e.g., DR) | ST | RE |
| 7 | Degree (e.g., MD) | IS | X |
| 8 | Source Table | IS | O |
| 9 | Assigning Authority | HD\_01 | C(R/X) |
| 10 | Name Type Code | ID | O |
| 11 | Identifier Check Digit | ST | O |
| 12 | Check Digit Scheme | ID | O |
| 13 | Identifier Type Code | ID | C(R/X) |
| 14 | Assigning Facility | HD | O |
| 15 | Name Representation Code | ID | O |
| 16 | Name Context | CE | O |
| 17 | Name Validity Range | DR\_DTM | O |
| 18 | Name Assembly Order | ID | O |
| 19 | Effective Date | TS | O |
| 20 | Expiration Date | TS | O |
| 21 | Professional Suffix | ST | O |
| 22 | Assigning Jurisdiction | CWE | O |
| 23 | Assigning Agency or Department | CWE | O |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XCN.13 | C(R/X) | If XCN.1 (ID Number) is valued |
| XCN.9 | C(R/X) | If XCN.1 (ID Number) is valued |

#### XCN\_02 - Extended Composite ID Number and Name for Persons

Versions: 2.5, 2.5.1

Support for Assigning Authority (XCN.9) and Identifier Type Code (XCN.13) is required. Support for other core components is required but allowed to be empty.

Use of this flavor is recommended when the person identifier need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | ID Number | ST | RE |
| 2 | Family Name | FN\_01 | RE |
| 3 | Given Name | ST | RE |
| 4 | Second and Further Given Names or Initials Thereof | ST | RE |
| 5 | Suffix (e.g., JR or III) | ST | RE |
| 6 | Prefix (e.g., DR) | ST | RE |
| 7 | Degree (e.g., MD) | IS | X |
| 8 | Source Table | IS | O |
| 9 | Assigning Authority | HD\_02 | C(R/X) |
| 10 | Name Type Code | ID | O |
| 11 | Identifier Check Digit | ST | O |
| 12 | Check Digit Scheme | ID | O |
| 13 | Identifier Type Code | ID | C(R/X) |
| 14 | Assigning Facility | HD | O |
| 15 | Name Representation Code | ID | O |
| 16 | Name Context | CE | O |
| 17 | Name Validity Range | DR\_DTM | O |
| 18 | Name Assembly Order | ID | O |
| 19 | Effective Date | TS | O |
| 20 | Expiration Date | TS | O |
| 21 | Professional Suffix | ST | O |
| 22 | Assigning Jurisdiction | CWE | O |
| 23 | Assigning Agency or Department | CWE | O |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XCN.13 | C(R/X) | If XCN.1 (ID Number) is valued |
| XCN.9 | C(R/X) | If XCN.1 (ID Number) is valued |

#### XCN\_03 - Extended Composite ID Number and Name for Persons

Versions: 2.6

Support for Assigning Authority (XCN.9) and Identifier Type Code (XCN.13) is required. Support for other core components is required but allowed to be empty.

Use of this flavor is recommended when the person identifier must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by requiring the Assigning Authority (XCN.9) when an ID is being exchanged and constraining it to require support for ISO complaint OIDs.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | ID Number | ST | RE |
| 2 | Family Name | FN\_01 | RE |
| 3 | Given Name | ST | RE |
| 4 | Second and Further Given Names or Initials Thereof | ST | RE |
| 5 | Suffix (e.g., JR or III) | ST | RE |
| 6 | Prefix (e.g., DR) | ST | RE |
| 7 | Degree (e.g., MD) | IS | X |
| 8 | Source Table | IS | O |
| 9 | Assigning Authority | HD\_05 | C(R/X) |
| 10 | Name Type Code | ID | O |
| 11 | Identifier Check Digit | ST | O |
| 12 | Check Digit Scheme | ID | O |
| 13 | Identifier Type Code | ID | C(R/X) |
| 14 | Assigning Facility | HD | O |
| 15 | Name Representation Code | ID | O |
| 16 | Name Context | CWE | O |
| 17 | Name Validity Range | DR\_DTM | O |
| 18 | Name Assembly Order | ID | O |
| 19 | Effective Date | DTM | O |
| 20 | Expiration Date | DTM | O |
| 21 | Professional Suffix | ST | O |
| 22 | Assigning Jurisdiction | CWE | O |
| 23 | Assigning Agency or Department | CWE | O |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XCN.13 | C(R/X) | If XCN.1 (ID Number) is valued |
| XCN.9 | C(R/X) | If XCN.1 (ID Number) is valued |

#### XCN\_04 - Extended Composite ID Number and Name for Persons

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for Person Identifier (XCN.1), Assigning Authority (XCN.9) and Identifier Type Code (XCN.13) is required. Support for other core components is required but allowed to be empty.

Use of this flavor is recommended when the person identifier need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Person Identifier | ST | C(R/RE) |
| 2 | Family Name | FN\_02 | RE |
| 3 | Given Name | ST | RE |
| 4 | Second and Further Given Names or Initials Thereof | ST | RE |
| 5 | Suffix (e.g., JR or III) | ST | O |
| 6 | Prefix (e.g., DR) | ST | O |
| 7 | Degree (e.g., MD) | - | X |
| 8 | Source Table | CWE | O |
| 9 | Assigning Authority | HD\_06 | C(R/X) |
| 10 | Name Type Code | ID | RE |
| 11 | Identifier Check Digit | ST | O |
| 12 | Check Digit Scheme | ID | O |
| 13 | Identifier Type Code | ID | C(R/X) |
| 14 | Assigning Facility | HD | O |
| 15 | Name Representation Code | ID | O |
| 16 | Name Context | CWE | O |
| 17 | Name Validity Range | - | X |
| 18 | Name Assembly Order | ID | O |
| 19 | Effective Date | DTM | O |
| 20 | Expiration Date | DTM | O |
| 21 | Professional Suffix | ST | O |
| 22 | Assigning Jurisdiction | CWE | C |
| 23 | Assigning Agency or Department | CWE | C |
| 24 | Security Check | ST | O |
| 25 | Security Check Scheme | ID | O |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XCN.13 | C(R/X) | If XCN.1 (Person Identifier) is valued |
| XCN.9 | C(R/X) | If XCN.1 (Person Identifier) is valued |
| XCN.1 | C(R/RE) | [If XCN.2 (Family Name) is not valued] AND [If XCN.3 (Given Name) is valued] |

### XON - Extended Composite Name and Identification Number for Organizations

The XON data type is used for specifying the name and/or ID number of an organization.

Best Practices for Constructing XON Data Type Flavors:

* The data type should be constructed so as to require an organization identifier or name in a given instance of the data type.

#### XON\_01 - Extended Composite Name and Identification Number for Organizations

Versions: 2.5, 2.5.1

Support for Assigning Authority (XON.6), Identifier Type Code (XON.7) and Organization Identifier (XON.10) is required. Support for Organization Name (XON.1) is required but allowed to be empty.

Use of this flavor is recommended when the organization identifier must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by requiring the Assigning Authority (XON.6) when an ID is being exchanged and constraining it to require support for ISO complaint OIDs.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Organization Name | ST | RE |
| 2 | Organization Name Type Code | IS | O |
| 3 | ID Number | NM | X |
| 4 | Check Digit | NM | O |
| 5 | Check Digit Scheme | ID | O |
| 6 | Assigning Authority | HD\_01 | C(R/X) |
| 7 | Identifier Type Code | ID | C(R/X) |
| 8 | Assigning Facility | HD | O |
| 9 | Name Representation Code | ID | O |
| 10 | Organization Identifier | ST | C(R/RE) |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XON.10 | C(R/RE) | If XON.1 (Organization Name) is not valued |
| XON.6 | C(R/X) | If XON.10 (Organization Identifier) is valued |
| XON.7 | C(R/X) | If XON.10 (Organization Identifier) is valued |

#### XON\_02 - Extended Composite Name and Identification Number for Organizations

Versions: 2.5, 2.5.1

Support for Assigning Authority (XON.6), Identifier Type Code (XON.7) and Organization Identifier (XON.10) is required. Support for Organization Name (XON.1) is required but allowed to be empty.

Use of this flavor is recommended when the organization identifier need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Organization Name | ST | RE |
| 2 | Organization Name Type Code | IS | O |
| 3 | ID Number | NM | X |
| 4 | Check Digit | NM | O |
| 5 | Check Digit Scheme | ID | O |
| 6 | Assigning Authority | HD\_02 | C(R/X) |
| 7 | Identifier Type Code | ID | C(R/X) |
| 8 | Assigning Facility | HD | O |
| 9 | Name Representation Code | ID | O |
| 10 | Organization Identifier | ST | C(R/RE) |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XON.10 | C(R/RE) | If XON.1 (Organization Name) is not valued |
| XON.6 | C(R/X) | If XON.10 (Organization Identifier) is valued |
| XON.7 | C(R/X) | If XON.10 (Organization Identifier) is valued |

#### XON\_03 - Extended Composite Name and Identification Number for Organizations

Versions: 2.5, 2.5.1

Support for Organization Name (XON.1), Assigning Authority (XON.6), Identifier Type Code (XON.7) and Organization Identifier (XON.10) is required.

Use of this flavor is recommended when the organization name is required and the organization identifier must be globally unique to enable broad interoperability across organizational and enterprise boundaries. Global uniqueness is achieved by requiring the Assigning Authority (XON.6) when an ID is being exchanged and constraining it to require support for ISO complaint OIDs.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Organization Name | ST | R |
| 2 | Organization Name Type Code | IS | O |
| 3 | ID Number | NM | X |
| 4 | Check Digit | NM | O |
| 5 | Check Digit Scheme | ID | O |
| 6 | Assigning Authority | HD\_01 | C(R/X) |
| 7 | Identifier Type Code | ID | C(R/X) |
| 8 | Assigning Facility | HD | O |
| 9 | Name Representation Code | ID | O |
| 10 | Organization Identifier | ST | C(R/RE) |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XON.10 | C(R/RE) | If XON.1 (Organization Name) is not valued |
| XON.6 | C(R/X) | If XON.10 (Organization Identifier) is valued |
| XON.7 | C(R/X) | If XON.10 (Organization Identifier) is valued |

#### XON\_04 - Extended Composite Name and Identification Number for Organizations

Versions: 2.5, 2.5.1

Support for Organization Name (XON.1) is required.

Use of this flavor is recommended when only the organization name is important.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Organization Name | ST | R |
| 2 | Organization Name Type Code | IS | X |
| 3 | ID Number | NM | X |
| 4 | Check Digit | NM | X |
| 5 | Check Digit Scheme | ID | X |
| 6 | Assigning Authority | HD\_01 | X |
| 7 | Identifier Type Code | ID | X |
| 8 | Assigning Facility | HD | X |
| 9 | Name Representation Code | ID | X |
| 10 | Organization Identifier | ST | X |

.

#### XON\_05 - Extended Composite Name and Identification Number for Organizations

Versions: 2.8, 2.8.1, 2.8.2

Support for Assigning Authority (XON.6), Identifier Type Code (XON.7) and Organization Identifier (XON.10) is required. Support for Organization Name (XON.1) is required but allowed to be empty.

Use of this flavor is recommended when the organization identifier need not be globally unique.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Organization Name | ST | RE |
| 2 | Organization Name Type Code | CWE | O |
| 3 | ID Number | - | X |
| 4 | Identifier Check Digit | - | X |
| 5 | Check Digit Scheme | - | X |
| 6 | Assigning Authority | HD\_06 | C(R/X) |
| 7 | Identifier Type Code | ID | C(R/X) |
| 8 | Assigning Facility | HD | O |
| 9 | Name Representation Code | ID | O |
| 10 | Organization Identifier | ST | C(R/RE) |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XON.10 | C(R/RE) | If XON.1 (Organization Name) is not valued |
| XON.6 | C(R/X) | If XON.10 (Organization Identifier) is valued |
| XON.7 | C(R/X) | If XON.10 (Organization Identifier) is valued |

### XPN - Extended Person Name

The XPN data type is used for specifying a person name plus associated information.

Best Practices for Constructing XPN Data Type Flavors:

* At least one component should be required, if only the Name Type Code (XPN.7).

Best Practices for Implementing XPN Data Type Flavors:

* Consider whether or not a person name will always be available. For example, an unconscious patient may be the subject of a message but unable to provide a name.
* Thought should be given to the provision of a variety of name types including an unknown name, a pseudonym or unassigned name (e.g. a baby who hasn’t been named yet).

#### XPN\_01 - Extended Person Name

Versions: 2.5, 2.5.1

Support for both the Name Type Code (XPN.7) is required. Support for the Family Name (XPN.1), Given Name (XPN.2), Second Name (XPN.3 and Suffix (XPN.4) is required but may be empty.

Use of this flavor is recommended when a name is not guaranteed to be known.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Family Name | FN\_01 | RE |
| 2 | Given Name | ST | RE |
| 3 | Second and Further Given Names or Initials Thereof | ST | RE |
| 4 | Suffix (e.g., JR or III) | ST | RE |
| 5 | Prefix (e.g., DR) | ST | O |
| 6 | Degree (e.g., MD) | IS | X |
| 7 | Name Type Code | ID | R |
| 8 | Name Representation Code | ID | O |
| 9 | Name Context | CE | O |
| 10 | Name Validity Range | DR\_DTM | X |
| 11 | Name Assembly Order | ID | O |
| 12 | Effective Date | TS | O |
| 13 | Expiration Date | TS | O |
| 14 | Professional Suffix | ST | O |

#### XPN\_02 - Extended Person Name

Versions: 2.5, 2.5.1

Support for both the Family Name (XPN.1), Given Name (XPN.2) and Name Type Code (XPN.7) is required. Support for the Second Name (XPN.3) and Suffix (XPN.4) is required but may be empty.

Use of this flavor is recommended when a name must be known.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Family Name | FN\_01 | R |
| 2 | Given Name | ST | C(R/X) |
| 3 | Second and Further Given Names or Initials Thereof | ST | RE |
| 4 | Suffix (e.g., JR or III) | ST | RE |
| 5 | Prefix (e.g., DR) | ST | O |
| 6 | Degree (e.g., MD) | IS | X |
| 7 | Name Type Code | ID | R |
| 8 | Name Representation Code | ID | O |
| 9 | Name Context | CE | O |
| 10 | Name Validity Range | DR\_DTM | X |
| 11 | Name Assembly Order | ID | O |
| 12 | Effective Date | TS | O |
| 13 | Expiration Date | TS | O |
| 14 | Professional Suffix | ST | O |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XPN.2 | C(R/X) | If the value of XPN.2 (Given Name) is not '""'. |

#### XPN\_03 - Extended Person Name

Versions: 2.5, 2.5.1

Support for both the Family Name (XPN.1) and Name Type Code (XPN.7) is required. Support for the Given Name (XPN.2), Second Name (XPN.3) and Suffix (XPN.4) is required but may be empty.

Use of this flavor is recommended when a family name must be known.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Family Name | FN\_01 | R |
| 2 | Given Name | ST | RE |
| 3 | Second and Further Given Names or Initials Thereof | ST | RE |
| 4 | Suffix (e.g., JR or III) | ST | RE |
| 5 | Prefix (e.g., DR) | ST | O |
| 6 | Degree (e.g., MD) | IS | X |
| 7 | Name Type Code | ID | R |
| 8 | Name Representation Code | ID | O |
| 9 | Name Context | CE | O |
| 10 | Name Validity Range | DR\_DTM | X |
| 11 | Name Assembly Order | ID | O |
| 12 | Effective Date | TS | O |
| 13 | Expiration Date | TS | O |
| 14 | Professional Suffix | ST | O |

#### XPN\_04 - Extended Person Name

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for both the Family Name (XPN.1), Given Name (XPN.2) and Name Type Code (XPN.7) is required. Support for the Second Name (XPN.3) is required but may be empty.

Use of this flavor is recommended when a name must be known.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Family Name | FN\_02 | R |
| 2 | Given Name | ST | R |
| 3 | Second and Further Given Names or Initials Thereof | ST | RE |
| 4 | Suffix (e.g., JR or III) | ST | O |
| 5 | Prefix (e.g., DR) | ST | O |
| 6 | Degree (e.g., MD) | - | X |
| 7 | Name Type Code | ID | R |
| 8 | Name Representation Code | ID | O |
| 9 | Name Context | CWE | O |
| 10 | Name Validity Range | - | X |
| 11 | Name Assembly Order | ID | O |
| 12 | Effective Date | DTM | O |
| 13 | Expiration Date | DTM | O |
| 14 | Professional Suffix | ST | O |
| 15 | Called By | ST | O |

#### XPN\_05 - Extended Person Name

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for both the Family Name (XPN.1) and Name Type Code (XPN.7) is required.

Use of this flavor is recommended when exchanging the Maiden (Last) Name. Name Type Code (XPN.7) is constrained to "M".

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Family Name | FN\_02 | R |
| 2 | Given Name | ST | O |
| 3 | Second and Further Given Names or Initials Thereof | ST | O |
| 4 | Suffix (e.g., JR or III) | ST | O |
| 5 | Prefix (e.g., DR) | ST | O |
| 6 | Degree (e.g., MD) | - | X |
| 7 | Name Type Code | ID | R |
| 8 | Name Representation Code | ID | O |
| 9 | Name Context | CWE | O |
| 10 | Name Validity Range | - | X |
| 11 | Name Assembly Order | ID | O |
| 12 | Effective Date | DTM | O |
| 13 | Expiration Date | DTM | O |
| 14 | Professional Suffix | ST | O |
| 15 | Called By | ST | O |

**Conformance Statements**

| **ID** | **Description** |
| --- | --- |
| XPN\_05-01 | The value of XPN-7 (Name Type Code) SHALL be 'M'. |

### XTN - Extended Telecommunication Number

The XTN data type is used for specifying telecommunication information. Within a single instance of the data type, Email Address (XPN.4), Local Number (XPN.7) and Unformatted Telephone Number (XPN.12) are all mutually exclusive, one must be populated, but not multiple.

Best Practices for Constructing XTN Data Type Flavors:

* Per the HL7 base standard, best practice requires the population of XTN.2 when XTN.4 or XTN.7 is populated.

#### XTN\_01 - Extended Telecommunication Number

Versions: 2.5, 2.5.1

Support for Telecommunication Use Code (XTN.2), Telecommunication Equipment Type (XTN.3), Email Address (XTN.4), Area/City Code (XTN.6) and Local Number (XTN.7) is required. Support for the Extension (XTN.8) is required but may be empty.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Telephone Number | ST | X |
| 2 | Telecommunication Use Code | ID | R |
| 3 | Telecommunication Equipment Type | ID | R |
| 4 | Email Address | ST | C(R/X) |
| 5 | Country Code | NM | O |
| 6 | Area/City Code | NM | C(R/X) |
| 7 | Local Number | NM | C(R/X) |
| 8 | Extension | NM | C(RE/X) |
| 9 | Any Text | ST | O |
| 10 | Extension Prefix | ST | O |
| 11 | Speed Dial Code | ST | O |
| 12 | Unformatted Telephone number | ST | X |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XTN.4 | C(R/X) | If the value of XTN.3 (Telecommunication Equipment Type) is one of list values: X400, Internet. |
| XTN.6 | C(R/X) | If the value of XTN.3 (Telecommunication Equipment Type) is one of list values: PH CP, SAT, FX, TDD. |
| XTN.7 | C(R/X) | If the value of XTN.3 (Telecommunication Equipment Type) is one of list values: PH, CP, SAT, FX, TDD. |
| XTN.8 | C(RE/X) | If the value of XTN.3 (Telecommunication Equipment Type) is one of list values: PH, CP, SAT, FX, TDD. |

#### XTN\_02 - Extended Telecommunication Number

Versions: 2.7, 2.7.1, 2.8, 2.8.1, 2.8.2

Support for Telecommunication Use Code (XTN.2), Telecommunication Equipment Type (XTN.3), Communication Address (XTN.4) and Local Number (XTN.7) is required. Support for the Area/City Code (XTN.6) is required but may be empty.

**Data Type Definition**

| **Seq** | **Element name** | **Data type** | **Usage** |
| --- | --- | --- | --- |
| 1 | Telephone Number | - | X |
| 2 | Telecommunication Use Code | ID | R |
| 3 | Telecommunication Equipment Type | ID | R |
| 4 | Communication Address | ST | C(R/X) |
| 5 | Country Code | SNM | O |
| 6 | Area/City Code | SNM | C(RE/X) |
| 7 | Local Number | SNM | C(R/X) |
| 8 | Extension | SNM | O |
| 9 | Any Text | ST | O |
| 10 | Extension Prefix | ST | O |
| 11 | Speed Dial Code | ST | O |
| 12 | Unformatted Telephone number | ST | X |
| 13 | Effective Start Date | DTM | O |
| 14 | Expiration Date | DTM | O |
| 15 | Expiration Reason | CWE | O |
| 16 | Protection Code | CWE | O |
| 17 | Shared Telecommunication Identifier | EI | O |
| 18 | Preference Order | NM | O |

**Conditional Predicates**

| **Location** | **Usage** | **Description** |
| --- | --- | --- |
| XTN.4 | C(R/X) | If the value of XTN.3 (Telecommunication Equipment Type) is one of list values: Internet, X.400. |
| XTN.6 | C(RE/X) | If the value of XTN.3 (Telecommunication Equipment Type) is not one of list values: Internet, X.400. |
| XTN.7 | C(R/X) | If the value of XTN.3 (Telecommunication Equipment Type) is not one of list values: Internet, X.400. |

# Appendix A – Base Standard Data Type Evolution

Table 10 provides a comparison chart that indicates when a base standard data type definition changed. Changes are indicated by a change in the number in the table. Zero (“0”) represent the origin. Note, for this analysis and for the standardized data type library, version 2.3.1 is the origin version. A grey box indicates that he data type for this version did not exist. The criteria for what constitute a change are given below:

* + number of components
  + name of component
  + data type of a component
  + usage
  + minimum length
  + maximum length
  + conformance length

This chart provides the analysis that determined the standardized data type flavor sets. For example, the initial version of the “CE” data type was defined in HL7 v2.3.1, modified in v2.4, and then again in v2.5. In v2.5.1, it remained the same, but was modified again in v2.6, and finally was deprecated in v2.7 and beyond. So, for “CE”, there are potentially four different SDTFs, and the same SDTF is specified for both v2.5 and v2.5.1. As is apparent, in almost all cases there are multiple SDTFs for each base data type definition, and each is associated with one or more versions of the HL7 v2 standard.

Table : Base Standard Data Type Evolution

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **HL7 v2 Version** | | | | | | | | | |
| **Data Type** | **2.3.1** | **2.4** | **2.5** | **2.5.1** | **2.6** | **2.7** | **2.7.1** | **2.8** | **2.8.1** | **2.8.2** |
| **AD** | **0** | **1** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **AUI** | **0** | **1** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **CCD** | **0** | **0** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **3** |
| **CCP** | **0** | **0** | **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** |
| **CD** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **CE** | **0** | **1** | **2** | **2** | **3** |  |  |  |  |  |
| **CF** | **0** | **1** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **CK** | **0** | **1** |  |  |  |  |  |  |  |  |
| **CN** | **0** | **1** |  |  |  |  |  |  |  |  |
| **CNE** | **0** | **1** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **CNN** |  | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **CNS** | **0** |  |  |  |  |  |  |  |  |  |
| **CP** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **CQ** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **CSU** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **CWE** | **0** | **1** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **CX** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **DDI** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **DIN** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **DLD** | **0** | **0** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **3** |
| **DLN** | **0** | **0** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **3** |
| **DLT** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **DR** | **0** | **0** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **3** |
| **DT** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **DTM** |  |  | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **DTN** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **ED** | **0** | **0** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **3** |
| **EI** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **EIP** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **ELD** | **0** | **1** | **2** | **2** | **3** |  |  |  |  |  |
| **ERL** |  |  | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **1** |
| **FC** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **FN** | **0** | **1** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **FT** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **GTS** |  |  | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **HD** | **0** | **0** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **3** |
| **ICD** |  |  | **0** | **0** | **1** | **2** | **2** | **2** | **2** | **2** |
| **ID** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **IS** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **JCC** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **LA1** | **0** | **1** | **2** | **2** | **3** | **4** | **4** |  |  |  |
| **LA2** | **0** | **1** | **2** | **2** | **3** | **4** | **4** |  |  |  |
| **MA** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **MO** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **MOC** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **MOP** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **MSG** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **NA** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **NDL** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **NM** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **NR** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **NUL** |  |  |  | **0** |  |  |  |  |  |  |
| **OCD** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **OG** |  |  |  |  |  |  |  |  |  | **0** |
| **OSD** | **0** | **0** | **1** | **1** | **1** |  |  |  |  |  |
| **OSP** | **0** | **1** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **PCF** | **0** | **0** |  |  |  |  |  |  |  |  |
| **PI** | **0** | **0** |  |  |  |  |  |  |  |  |
| **PIP** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **PL** | **0** | **0** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **3** |
| **PLN** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **PN** | **0** | **1** |  |  |  |  |  |  |  |  |
| **PPN** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **PRL** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **PT** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **PTA** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **QIP** | **0** | **1** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **QSC** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **RCD** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **RFR** | **0** | **1** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **RI** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **RMC** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **RP** | **0** | **0** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **3** |
| **RPT** |  |  | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **1** |
| **SAD** |  | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **SCV** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **SI** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **SN** | **0** | **1** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **SNM** |  |  |  |  |  | **0** | **0** | **0** | **0** | **0** |
| **SPD** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **SPS** | **0** | **1** | **2** | **2** | **2** |  |  |  |  |  |
| **SRT** |  | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **ST** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **TM** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **TN** | **0** | **0** |  |  |  |  |  |  |  |  |
| **TQ** | **0** | **1** | **2** | **2** | **3** |  |  |  |  |  |
| **TS** | **0** | **0** | **1** | **1** | **2** |  |  |  |  |  |
| **TX** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **UVC** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **VARIES** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **varies** | **0** | **0** | **0** | **0** | **0** | **0** |  |  |  |  |
| **VH** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **VID** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **VR** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **WVI** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **WVS** | **0** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** |
| **XAD** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **XCN** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **XON** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **5** | **5** | **5** |
| **XPN** | **0** | **1** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **XTN** | **0** | **0** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **3** |

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Data Type Flavor** | Is a specialization of a base data type flavor. That is, a data type defined in the base HL7 v2.x standard is constrained for a particular use, that constrained data type is given an identifier and is deemed a data type flavor. |
| **DTF** | Data Type Flavor – Is the generic term to include both SDTF (Standard Data Type Flavor) and UDTF (User Data Type Flavor). |
| **Implementation Guide** | Is a specification that is created to organize a collection of message profiles for specifying a set of related interactions described in a use case or use cases. |
| **Profiling** | The process of placing additional constraints on a message definition and any of its constitute elements. |
| **SDTF** | Standard Data Type Flavor – Is a Data Type Flavor created and managed by HL7 and is the central focus of this document. |
| **Specifier** | A person who is creating the implementation guide or message profile. |
| **UDTF** | User Data Type Flavor – Is a Data Type Flavor created and managed by the user of the HL7 v2.x standard. This document provides creation and management guidelines and naming conventions for UDTFs. |

1. Except in some recent guides. [↑](#footnote-ref-1)