

work test Release 1.6 try 2

Subtitle Default Sub Title

May 24, 2016

HL7 Version 2.5.1

Document Version null

NIST

# Introduction

Immunization Information Systems (IIS) are centralized population based repositories of immunization related information. They receive and share data on individual clients/patients (note that client and patient are terms which are used interchangeably in this document) with a number of other systems, including Electronic Health Record systems (EHR-S). Health Level Seven (HL7) is a nationally recognized standard for electronic data exchange between systems housing health care data. The HL7 standard is a key factor that supports this two-way exchange of information because it defines a syntax or grammar for formulating the messages that carry this information. It further describes a standard vocabulary that is used in these messages. It does not depend on specific software, that is, it is platform independent.

This document represents the collaborative effort of the American Immunization Registry Association (AIRA) and the Centers for Disease Control and Prevention (CDC) to improve inter-system communication of immunization records. The effort has received input from the National Institute of Standards and Technology (NIST) to improve the capacity to test conformance with this Implementation Guide. In addition, this Guide addresses a need to specify usage requirements for data elements that are not included in the standard HL7 usage designations. This implementation guide replaces the Implementation Guide for Immunization Data Transaction Using Version 2.3.1 of the HL7 Standard Protocol, and previous versions of this Guide. It is based on HL7 Version 2.5.1, as published by the HL7 organization (www.hl7.org). In addition, it pre-adopts a number of features of HL7 Version 2.7.1, such as data types and the conformance model (defined in Chapter 2B).

As HL7 has developed and published new versions of the standard, it has sought to maximize the ability of implementations, based on newer versions to be able to accept messages from earlier versions. Based on this, we anticipate that faithful implementations of this Guide will be able to accept most immunization messages based on the 2.3.1 Guide. Note that variations in current 2.3.1 interfaces increase the risk that faithful 2.5.1 implementations will encounter problems with 2.3.1 messages.

Implementations that are supporting Version 2.3.1 messages should continue to follow the specifications of 2.3.1 messages described in the Implementation Guide Version 2.2, June 2006.

## Purpose

While this document is intended to be applicable across a large range of systems and jurisdictions, local policies and requirements may necessitate additional constraints. One way to insure success in such a scenario is to publish a local profile or implementation guide that outlines the local business rules and processes. These guides may further constrain this Guide, but may not contradict it.

This Guide makes the following assumptions:

* Infrastructure is in place to allow accurate and secure information exchange between information systems.
* Providers access immunization information through either an EHR-S or immunization information system (IIS).
* Privacy and security has been implemented at an appropriate level.
* The immunization record and demographic record for each patient contains sufficient information for the sending system to construct the immunization and demographic message properly.
* External business rules are documented locally.

Note that the focus of this guide is on the format and grammar of the messages between systems. The activities shown within a system are intended to put the message in context and to highlight the local responsibilities for successful messaging.

## Audience

This Guide has two audiences. The first is the system managers that must understand this process at a high level. The second is the technical group from IIS and EHR-S that must implement these guidelines. For them we strive for an unambiguous specification for creating and interpreting messages. Our goal is for this Guide to be a bridge between the two.

It is important to note that HL7 specifies the exchange between 2 systems. It does not specify how any given system is implemented to accomplish the goals of messaging.

## Organization of this guide

The introductory chapters are meant to lay out what can be done and why. The chapters that follow them describe and specify how. This document defines message profiles which are used as the basis for building and consuming individual messages during an exchange. Message profiles are combined to define message exchanges which will support the major use cases.

## Referenced profiles - antecedents

None

## Scope

### In Scope

This Guide is intended to facilitate the exchange of immunization records between different systems, including IIS, EHR-S and other health data systems. This includes:

* sending immunization event histories for a patient
* acknowledging receipt of immunization event histories
* requesting a complete immunization history for a patient
* responding to a request for a complete immunization history
* requesting an evaluated history and forecast for a patient
* responding to requests for an evaluated history and forecast
* reporting errors in the messaging process
* sending observations about a patient or immunization event (this may include patient eligibility for a funding program, reactions, forecasts and evaluations).

### Out of Scope

The Guide is not intended to address other issues such as:

* business rules, which are not implicit in HL7, applied when creating a message, including defining events which trigger the creation of the message
* business rules, which are not implicit in HL7, applied when processing a received message
* a standard transport layer
* search process used when responding to a query
* business rules used to deduplicate clients or events
* management of vaccine inventory
* legal and governance issues regarding data access authorizations, data ownership and data use
* maintenance of Master Person Index (MPI).

Local implementations are responsible for the important issues described above.

#### **new Section48211**

sidojfodsakjgojdef;lkgbjmfkgpeakp[

## Key technical decisions [conventions]

**HL7 definitions**

The terms below are organized to move from the message to subsequently more granular components.

**Message**: A message is the entire unit of data transferred between systems in a single transmission. It is a series of segments in a sequence defined by the message specifications. These specifications are based on constraints to the HL7 specifications, as described in an Implementation Guide.

**Segment Group**: A segment group is a logical collection of segments. Segment groups defined within a message may be required or optional, may occur only once or may be allowed to repeat.

**Segment**: A segment is a logical grouping of data fields. Segments within a defined message may be required or optional, may occur only once, or may be allowed to repeat. Each segment is named and is identified by a segment ID, a unique 3-character code.

**Field**: A field is a string of characters and is of a specific data type. Each field is identified by the segment it is in and its position within the segment; e.g., PID-5 is the fifth field of the PID segment. A field is preceded by the | character.

**Component**: A component is one of a logical grouping of items that comprise the contents of a coded or composite field. Within a field having several components, not all components are required to be valued.

**Data type**: A data type restricts the contents and format of the data field. Some data types are coded or composite types with several components. The applicable data type is listed and defined in each field definition.

**Code Sets/Systems and Value Sets**: Most data elements will have associated lists of acceptable values in tables supported by a standards organization such as HL7 or CDC. These code sets will include definitions to support common usage. This document references both Coding Systems (AKA Code Sets) and Value Sets in relation to coded message elements (including the CE, CWE, CX, IS, ID and XCN data types). While related, these terms are distinct. A Coding System is an extensive, and in some cases extendable, list of values available for use in a message. A single Coding System may be relevant to a number of different parts of a single message. For example, HL7 table 0203 contains a list of Identifier Types. This table is called out as part of the CX data type (used in PID-3 and QPD-3) as well as the XCN data type (used in ORC-12 and RXA-10). A Coding System tends to be a very broad list and not all values are appropriate to use in a given message element. For example HL7 table 0203 contains the ID types of MR (Medical Record Number) and NPI (National Provider Identifier) which are appropriate for use in PID-3 and ORC-12 respectively. In contrast, a Value Set is a more refined list of values, taken from one or more Coding Systems, applied at a more granular level of the message and which contains only values appropriate for that location in the message. In some cases, a Value Set may have the same content as the underlying Coding System.

**Delimiters**: Delimiter characters are used to separate segments, fields and components in an HL7 message. The delimiter values are given in MSH-2 and used throughout the message. Applications must use agreed upon delimiters to parse the message. Messages used in this Guide SHALL use the following delimiters:  
<CR> = Segment Terminator;  
| = Field Separator;  
^ = Component Separator;  
& = Sub-Component Separator;  
~ = Repetition Separator;  
\ = Escape Character.

**Message syntax**: Each message is defined in special notation that lists the segment 3-letter identifiers in the order they will appear in the message. Braces, {}, indicate that one or more of the enclosed group of segments may repeat, and brackets, [ ], indicate that the enclosed group of segments is optional. Note that segments may be nested within the braces and brackets. This will indicate that the nested segments are units within a subgroup of segments. Their Usage is relative to the parent segment in the group.

**Z Segments**: All message types, trigger event codes, and segment ID codes beginning with Z are reserved for locally defined messages. No such codes will be defined within the HL7 Standard. The users of this Guide have agreed to eliminate Z segments from their implementations in order to produce a standard method that will be used nationally to transmit immunization data. Even though the profile identifier used by this this implementation guide (Z22, Z23, etc) begin with Z they are not locally defined and they are not Z-segments.

**Pre-Adoption Of Some Features Of HL7 Version 2.7.1**

This implementation Guide pre-adopts some features of HL7 Version 2.7.1 to support improved consistency in implementation with the goal of improving interoperability. These include:

* Conformance statements
* Conditional predicates
* Usage guidance
* New fields in MSH segment

**Use of Vocabulary Standards**

This guide calls for specific vocabulary standards for the exchange of immunization information such as LOINC and SNOMED. Standard vocabularies enable automated decision support for patient healthcare, as well as for public health surveillance of populations. Terminology is updated periodically and it is best practice to use the most current version of the coding system.

**Conventions**

This guide adheres to the following conventions:

* The guide is constructed assuming the implementer has access to the Version 2.5.1 of the HL7 Standard. Although some information from the standard is included in this implementation guide, much information from the standard has not been repeated here.
* The rules outlined in HL7 2.7.1, Chapter 2B, Section 2B, Conformance Using Message Profiles, were used to document the use case for, and constraints applied to, the messages described in this guide
* Data types have been described separately from the fields that use the data types.
* No conformance information is provided for optional message elements. This includes length, usage, cardinality, value sets and descriptive information. Implementers who want to use optional message elements should refer to the base HL7 V2.5.1 Standard to determine how these optional message elements will be used.
* This guide uses X as a conformance usage indicator very sparingly. Where the underlying standard indicates the segments/field/component is present for backwards compatibility (B) or withdrawn ("W") an X will be used. Some conditional elements may have a usage of X if the predicate condition is the only case where the element is used. For all other fields/components O is used to enable trading partners to explore additional capabilities. Note that without a clearly agreed to complementary profile between trading partners, a sender does not have to send any elements marked as an "O", nor does a receiver have to process any elements marked as an "O".

**Null and empty fields**

The delete indicator is transmitted as two double quote marks (). A delete indicator valued field differs from an empty field. An empty field should not overwrite previously sent data, while the delete indicator value means that any previous value should be deleted.

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| "" | Delete the value recorded in the receiving system data base. |
| <empty> | Make no changes to the record in the receiving data base. The sending system has no information on this field. |

The delete indicator () should not be sent in immunization messages. Systems which will send the delete indicator () must specify their use in local implementation guides. Systems which will accept and process the delete indicator, as described above, must specify their use in local implementation guides.

**Keywords**

The following keywords in this document are to be interpreted as follows:

**MUST**or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.  
**MUST NOT**or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.  
**SHOULD**or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.  
**SHOULD NOT** or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.  
**MAY**or the adjective "OPTIONAL", mean that an item is truly optional. One software supplier may choose to include the item to enable certain capabilities while another software supplier may omit the same item. In either case, the communication partner cannot be expected to either provide it (sender) or process it (receiver) without clear and voluntary agreement between the partners.

An implementation which does not include a particular segment/field/component marked as optional MUST be prepared to interoperate with another implementation which does include the optional segment/field/component, though perhaps with reduced functionality. In the same vein an implementation which includes a particular segment/field/component marked as optional MUST be prepared to interoperate with another implementation which does not include the optional segment/field/component.

**Usage of Segments, Fields and Components**

Many fields and segments in HL7 are optional. This guide tightens constraints on some fields to support functionality required for meaningful use of immunization data. The following lists the rules applied to the decisions used to determine usage in this Guide.

1. Any segment, field, or component that is required by HL7 standard is required or required but may be empty.
2. Any field or component that is a required National Vaccine Advisory Committee (NVAC) Core Data element is required or required but may be empty .
3. Any segment that contains a required NVAC Core data element is required but may be empty.
4. Any segment, field, or component that is retained for backward compatibility in Version 2.5.1 SHALL be unsupported in this Guide.
5. Any segment, field, or component that is conditional but may be empty in Version 2.5.1 shall be conditional or conditional but may be empty in this Guide, unless this conflicts with 2 or 3 above.
6. All other fields will be left optional.

**Usage Conformance Testing Recommendations**

The following text is pre-adopted from the HL7 V2.7.1 Conformance (Chapter 2B) Please refer to the base standard documentation for a full explanation of conformance concepts. Usage is described here as it introduces the revised approach to conditional element handling.

**Usage**  
Message content is governed by the cardinality specification associated (explicitly or implicitly) with each element of an HL7 message. Usage rules govern the expected behavior of the sending application and receiving application with respect to the element. The usage codes expand/clarify the optionality codes defined in the HL7 standard. Usage codes are employed in a message profile to constrain the use of elements defined in the standard. The usage code definitions are given from a sender and receiver perspective and specify implementation and operational requirements.

The standard allows broad flexibility for the message structures that HL7 applications must be able to receive without failing. But while the standard allows that messages may be missing data elements or may contain extra data elements, it should not be inferred from this requirement that such messages are conformant. In fact, the usage codes specified in a message profile place strict conformance requirements on the behavior of the application.

**Definition Of Conditional Usage**  
C(a/b) - a and b in the expression are placeholders for usage codes representing the true (a) predicate outcome and the false (b) predicate outcome of the condition. The condition is expressed by a conditional predicate associated with the element (See the Error section in V2.7.1 Chapter 2B). a and b shall be one of R, RE, O and/or X. The values of a and b can be the same.

The example C(R/RE) is interpreted as follows. If the condition predicate associated with the element is true then the usage for the element is R-Required. If the condition predicate associated with the element is false then the usage for the element is RE- Required but may be empty.

There are cases where it is appropriate to value a and b the same. For example, the base standard defines the usage of an element as C and the condition predicate is dependent on the presence or non-presence of another element. The profile may constrain the element that the condition is dependent on to X; in such a case the condition should always evaluate to false. Therefore, the condition is profiled to C(X/X) since the desired effect is for the element to be not supported. Note it is not appropriate to profile the element to X since this breaks the rules of allowable usage profiling (see in V2.7.1 Chapter 2B table HL7 Optionality and Conformance Usage).

**Sending And Receiving Application Conformance Requirements**

**Table 3-2 Sending Application Conformance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Definition** | **Implementation Requirement** | **Operation Requirement** |
| R | Required | The application **SHALL**implement R elements. | The application **SHALL**populate R elements with a non-empty value. |
| RE | Required but may be empty | The application **SHALL**implement RE elements. | The application SHALL populate RE elements with a non-empty value if there is relevant data. The term relevant has a confounding interpretation in this definition (There are multiple interpretations of RE when a value is known. One is the capability must always be supported and a value is sent if known, the other is the capability must always be supported and a value may or may not be sent even when known based on a condition external to the profile specification. The condition may be noted in the profile but cannot be processed automatically. This is what can be interpreted from the relevant part of the definition. Regardless of the interpretation the RE usage code, a set of test circumstances can be developed to sufficiently test the RE element. See the Conformity Assessment of Conformance Constructs section for more details.) |
| C (a/b) | Conditional | An element with a conditional usage code has an associated condition predicate that determines the operational requirements (usage code) of the element.If the condition predicate associated with the element is true, follow the rules for a which shall be one of R, RE, O or X:  If the condition predicate associated with the element is false, follow the rules for b which shall be one of R, RE, O or X.a and b can be valued the same.  Note: when C(O/X) or similar is used a condition predicate will not be provided. | |
| X | Not supported in this guide | The application (or as configured) **SHALL NOT** implement X elements. | The application **SHALL NOT** populate X elements. |
| O | Optional | None. The usage indicator for this element has not yet been defined. For an implementation profile all optional elements must be profiled to R, RE, C(a/b), or X. | Not Applicable |

Note: Implementation Requirement the capability of the system. The Operation Requirement indicates what is included in the message.

Table 3-3 Receiving Application Conformance

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Definition** | **Implementation Requirement** | **Operation Requirement** |
| R | Required | The application **SHALL**implement R elements. | The receiving application **SHALL**process (save/print/archive/etc.) the information conveyed by a required element. (Processing does not necessarily require permanent storage of the required element. For instance OBX-4 (sub-id) is used to group associated OBX segments, but will probably not be stored.)  A receiving application **SHALL**raise an exception due to the absence of a required element. A receiving application **SHALL NOT**raise an error due to the presence of a required element. |
| RE | Required but may be empty | The application **SHALL**implement RE elements. | The receiving application **SHALL**process (save/print/archive/etc.) the information conveyed by a required but may be empty element. The receiving application SHALL process the message if the element is omitted (that is, an exception**SHALL NOT** be raised because the element is missing). |
| C (a/b) | Conditional | The usage code has an associated condition predicate that determines the operational requirements (usage code) of the element.  If the condition predicate associated with the element is true, follow the rules for a which **SHALL**be one of R, RE, O or X  If the condition predicate associated with the element is false, follow the rules for b which **SHALL**be one of R, RE, O or X.  a and b can be the same.  Note: when C(O/X) or similar is used a condition predicate will not be provided. | |
| X | Not supported in this guide | The application (or as configured) **SHALL NOT** implement X elements. | None, if the element is not sent.  If the element is sent the receiving application may process the message, **SHALL**ignore the element, and **MAY**raise an exception. The receiving application **SHALL NOT**process (save/print/archive/etc.) the information conveyed by a not-supported element. |
| O | Optional | None. The usage indicator for this element has not yet been defined. For an implementation profile all optional elements must be profiled to R, RE, C(a/b), or X. | None |

Note that local implementations may constrain the requirements of this Implementation Guide.  
These guides should indicate that an element that will be ignored by the local application should use symbol IX (Element will be ignored) to indicate that the element will be ignored by the local application. The Operation Requirement will be: If the element is not sent, no action is taken. If the element is sent, it is ignored.

**Encoding Rules for Sending**

1. Encode each segment in the order specified in the abstract message format.
2. Place the Segment ID first in the segment.
3. Precede each data field with the field separator.
4. Encode the data fields in the order and data type specified in the segment definition table.
5. End each segment with the segment terminator (carriage return).
6. Components, subcomponents, or repetitions that are not valued at the end of a field need not be represented by component separators. The data fields below, for example, are equivalent:  
   |^XXX&YYY&&^| is equal to |^XXX&YYY^|  
   |ABC^DEF^^| is equal to |ABC^DEF|
7. Components, subcomponents, or repetitions that are not valued, but precede components, subcomponents or occurrence that are valued must be represented by appropriate separators. For example, the following CE data type element has the first triplicate empty and a populated second triplicate:  
   |^^^ABC^Text^Codesystem|
8. If a field allows repetition (Cardinality maximum > 1), then the length of the field applies to EACH repetition.
9. No field separator is required after the last required field unless other following fields have data. The presence of a field separator after the last required field is NOT an error and may be ignored.

**Basic Message Processing Rules**

**Message Acknowledgement**

Original Mode processing is supported by this Implementation Guide. Enhanced Mode Acknowledgement is not supported.

The conversation between a sending system and a receiving system consists of a message (VXU, QBP) and a response (ACK, RSP). Receiving systems are expected to process the message and send a response. The system receiving the acknowledgement response does not acknowledge the response. In other words, the system receiving a VXU is expected to return an ACK. The system receiving that ACK is not expected to respond back to that ACK. All response messages (profiles Z23, Z32, Z31, Z33 and Z42) shall be returned synchronously. That is, the receiving process gives the response immediately or in a short period during which the originating process will wait for the response. The originating process will not send a new message until a response has been received. The originating process will not send a new message until a response has been received. A system may initiate multiple simultaneous processes if allowed, but each process must wait for a response to a given message before sending the next one. For query interactions, this behavior is controlled by the constrained value of I in the Query Priority (RCP-2) field. See Chapter 5 of the HL7 2.5.1 Base Standard for more details.

Profiles Z22, Z34 and Z44 have constrained values of ER for Accept Acknowledgement Type (MSH-15) and AL for Application Acknowledgement Type (MSH-16). When processing a message conformant with one of these profiles, the receiving system shall evaluate the message for unsupported message types, version ID, and processing IDs or other issues unrelated to format or content. If the message fails this validation, an ACK message conforming to profile Z23 shall be returned. This is consistent with ER in MSH-15. Messages which pass this initial validation are then processed and an appropriate Application level response message is returned. This is consistent with AL in MSH-16. Note that messages which fail the initial validation are not processed further and therefore do not have the opportunity to trigger an Application level response message. The receiving system only returns one message per incoming message. This process is diagrammed in Figure 37 (Z22 profile), Figure 41 (Z34 profile) and Figure 44 (Z44 profile).

Receipt and processing of ACK messages has a number of significant benefits:

* Notification of errors and rejected data alerts sender that message has errors and may require correction
* Alerting sending user that the data did not get into the receivers system

Some messages pass through intermediary systems like a Health Information Exchange (HIE). It is important that the intermediary system pass the ACK back to the sending system to allow the sending system to be aware of and deal with messaging errors.

**Processing Rules for Receiving System:**

In general, as much of a message should be consumed as possible. For example, a message or order group should be process even if lacks a non-required element. Similarly, if one order group cannot processed, remaining order groups should be processed if possible. The following table contains the rules for processing received messages. Note that these outcomes may cascade. That is, if a required field has bad data, it is empty. If that required field is empty, the segment is treated as empty. It that segment is not a part of a segment group and is empty, the message is rejected.

**Table 3-1 Receiving System Processing Rules**

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Outcome** | **Acknowledgement** | **Action** |
| Data fields are populated after last required field in segment. | Ignore the extra fields | No Error | Continue processing message |
| Data field violates data type specifications or contains unacceptable data. | Treat the field as empty | Send Error | Continue processing message |
| Required data field is empty. | Treat the segment as empty | Send Error | Continue processing message |
| Required But May Be Empty field is empty. | No outcome | No Error | Continue processing message |
| Required or conditionally required segment is empty or missing. | All data fields in segment are not present | Send Error | Continue processing message |
| Optional segment or unexpected segment is included. | Ignore the segment, this is not an error | No Error | Continue processing message |
| Data segment out of proper order. | Treat segment as empty | Send Error | Continue processing message |
| Required segment that is not part of a segment group is empty. | Reject message | Send Error | Reject message |
| Required segment that is part of a segment group is empty or missing. | Treat segment group as empty | Send Error | Continue processing message |
| Required segment group is empty or missing. | Reject message | Send Error | Reject message |

# Use Cases

## Actors

## General Assumptions

### Assumptions

### Pre-conditions

### Post-conditions

## Use Case

### User Story

### Specific Assumptions

#### **Assumptions**

#### **Pre-conditions**

#### **Post-conditions**

### Scenario

### Context

### Interaction Model

### Functional Requirements

# Message Infrastructure

## Conformance Profiles

## Segments and Field Descriptions

## Datatypes

## Value Sets