2. Control

Chapter Chairs/Editors: Mark Shafarman
Oracle

Larry Reis
Consultant

Mark Tucker
Regenstrief Institute for Health Care

Additional Editors: Mike Henderson
Kaiser Permanente

Joann Larson
Kaiser Permanente

2.1 CHAPTER 2 CONTENTS

2.2 INTRODUCTION

The Control chapter of this Standard defines the generic rules that apply to all messages. Subsequent sections define functionally specific messages to be exchanged among certain applications. The specific aspects of message definition that are addressed herein are:

a) the form to be used in functional chapters for describing messages. This includes their purpose, their contents, and the interrelationships among them. This form is called an abstract message definition because it is purely a level 7 (application) definition.

b) the HL7 encoding rules for converting an abstract message into a string of characters that comprises an actual message

c) the programming procedures required to exchange messages using the HL7 specifications

d) the anticipated relationship with lower level protocols

e) certain message segments that are components of all messages

f) a single message, the acknowledgment message, that may be used unchanged in multiple applications.

2.3 CONCEPTUAL APPROACH

2.3.1 Trigger events

The Standard is written from the assumption that an event in the real world of healthcare creates the need for data to flow among systems. The real-world event is called the **trigger event**. For example, the trigger event **a patient is admitted** may cause the need for data about that patient to be sent to a number of other systems. The trigger event, **an observation (e.g., a CBC result) for a patient is available,** may cause the need for that observation to be sent to a number of other systems. When the transfer of information is initiated by the application system that deals with the triggering event, the transaction is termed an **unsolicited update**.

Note:	No assumption is made about the design or architecture of the application system creating the unsolicited update. The scope of HL7 is restricted to the specification of messages between application systems and the events triggering them.

HL7 allows the use of trigger events at several different levels of data granularity and inter-relationships. For example, most Patient Administration (ADT) trigger events concern single objects (such as an admit event, which creates a message that contains data about a single person and/or account). Other ADT trigger events are concerned with relationships between more than one object (e.g., the merge events, which specify patient or account merges). Some ADT trigger events pertain to a collection of objects that may have no significant inter-relationships (e.g., a record-oriented location-based query, whose response contains data about a collection of inpatients who are related only temporarily by local geography).

2.3.2 Acknowledgments: original mode

When the unsolicited update is sent from one system to another, this acknowledgment mode specifies that it be acknowledged at the application level. The reasoning is that it is not sufficient to know that the underlying communications system guaranteed delivery of the message. It is also necessary to know that the receiving application processed the data successfully at a logical application level.

The acknowledgment may contain data of interest to the system that initiated the exchange. For example, if a patient care system has processed the trigger event **a lab test is ordered for a patient**, it may send an unsolicited update to a lab application identifying the patient, the test ordered, and various other information about the order. The ancillary system will acknowledge the order when it has processed it successfully. For some pairings of patient care and ancillary department systems the acknowledgment may also include the ancillary identification number that was assigned. (HL7 does not require Order Entry and Results Reporting applications to interface in this manner, but it supports those that do.)

The HL7 Standard makes no assumptions about the ownership of data. It also makes no requirements of its own on the subsequent action of the recipient of data, nor does it make any assumption about the design or architecture of the receiving application system. The scope of HL7 is restricted to the specification of messages between application systems, and the events triggering them. HL7 does not explicitly support, but can be used with, systems that support store and forward and data broadcast facilities (see the HL7 Implementation Support Guide).

The HL7 Standard makes no functional interpretation of the requirement that a system commit the data in a message to its database before acknowledging it. All that is required is that the receiving system accept responsibility for the data, providing the same integrity test that it would apply to data from any source. To continue the prior example, the ancillary system may acknowledge the order after placing it in an input queue, expecting to fully process the order into its database at a future time. The only assumption is that the input queue is maintained at the same level of integrity as the database.

2.3.3 Acknowledgments: enhanced mode

The HL7 acknowledgment paradigm has been extended to distinguish both accept and application acknowledgments, as well the conditions under which each is required. With a positive accept acknowledgment, the receiving system commits the message to safe storage in a manner that releases the sending system from the need to resend the message. After the message has been processed by the receiving system, an application acknowledgment may be used to return the resultant status to the sending system.

2.3.4 Queries

Query documentation including messages, segments, special protocols, implementation considerations and examples have been moved to chapter 5. The unsolicited display messages were also moved because their message syntax is query-like in nature.

2.4 COMMUNICATIONS ENVIRONMENT

The HL7 Standard defines the messages as they are exchanged among application entities and the procedures used to exchange them. As such, it conceptually operates at the seventh level of the ISO model for Open System Interconnection (OSI). It is primarily concerned with the data content and interrelationship of messages and with communicating certain application-level error conditions.

Since the OSI protocols are not universally implemented, the HL7 Working Group is interested in providing standards that will be useful in the interim. It is also recognized that there is now, and will continue to be, interest in communicating health data among systems operating in communications environments that provide a high level of functionality, but use protocols other than ISO OSI. The universe of environments of interest to HL7 includes, but is not restricted to:

a) ad hoc environments that do not provide even basic transport reliability. Such environments consist of point-to-point RS-232 links, modems, and even LANs, if their connection to host computers is made via RS-232 communications links. Until OSI high level standards become truly prevalent, many healthcare interfaces will be implemented over such links. In such an environment, the HL7 Lower Level Protocols (LLP) may be used between systems to enhance the capabilities of the communications environment. The HL7 Lower Level Protocols are defined in the HL7 Implementation Guide, which is not an official part of the Standard.

b) environments that support a robust transport level, but do not meet the high level requirements. This includes environments such as TCP/IP, DECNET, and SNA.

c) ISO and proprietary networks that implement up to presentation and other high level services. IBM's SNA LU6.2 and SUN Microsystems's NFS are examples of complete proprietary networks.

d) two or more applications running on the same physical and/or logical machine that are not tightly integrated. In these environments, the messaging capabilities may be provided by inter-process communications services (e.g., Pipes in a UNIX System).

The HL7 Standard assumes that the communications environment will provide the following capabilities:

a) error free transmission. Applications can assume that they correctly received all of the transmitted bytes in the order in which they were sent. This implies that error checking is done at a lower level. However, sending applications may not assume that the message was actually received without receiving an acknowledgment message.

b) character conversion. If the two machines exchanging data use different representations of the same character set, the communications environment will convert the data from one representation to the other.

c) message length. HL7 sets no limits on the maximum size of HL7 messages. The Standard assumes that the communications environment can transport messages of any length that might be necessary. In practice, sites may agree to place some upper bound on the size of messages and may use the message continuation protocol, described later in this chapter, for messages that exceed the upper limit.

Note:	Just as HL7 makes no assumptions about the design or architecture of the application systems sending and receiving HL7 messages, it makes no assumptions about the communications environment beyond those listed above. In particular, aside from the above assumptions, the communications environment, including its architecture, design and implementation, is outside the scope of HL7.

2.5 HL7 MESSAGES

This section and Sections 2.6, "SEGMENTS," through 2.10, "Use of escape sequences in text fields," define the components of messages and provide the methodology for defining abstract messages that are used in later chapters. A **message** is the atomic unit of data transferred between systems. It is comprised of a group of segments in a defined sequence. Each message has a **message type** that defines its purpose. For example the ADT Message type is used to transmit portions of a patient's Patient Administration (ADT) data from one system to another. A three-character code contained within each message identifies its type. These are listed in the Message Type list, Appendix A.

The real-world event that initiates an exchange of messages is called a trigger event. (See Section 2.3.1, "Trigger events," for a more detailed description of trigger events.) Appendix A contains the codes that represent all defined trigger events. These codes represent values such as **A patient is admitted** or **An order event occurred**. There is a one-to-many relationship between message types and trigger event codes. The same trigger event code may not be associated with more than one message type; however a message type may be associated with more than one trigger event.

All message types and trigger event codes beginning with the letter "Z" are reserved for locally-defined messages. No such codes will be defined within the HL7 Standard.

2.6 SEGMENTS

A **segment** is a logical grouping of **data fields**. Segments of a message may be required or optional. They may occur only once in a message or they may be allowed to repeat. Each segment is given a name. For example, the ADT message may contain the following segments: Message Header (MSH), Event Type (EVN), Patient ID (PID), and Patient Visit (PV1).

Each segment is identified by a unique three-character code known as the Segment ID. Although the actual segments are defined in various chapters, the ID codes assigned to the various segments are listed in Appendix A.

All segment ID codes beginning with the letter **Z** are reserved for locally-defined messages. No such codes will be defined within the HL7 Standard.

2.7 FIELDS

Definition: A field is a string of characters.

HL7 does not care how systems actually store data within an application. When fields are transmitted, they are sent as character strings. Except where noted, HL7 data fields may take on the null value. Sending the null value, which is transmitted as two double quote marks (""), is different from omitting an optional data field. The difference appears when the contents of a message will be used to update a record in a database rather than create a new one. If no value is sent, (i.e., it is omitted) the old value should remain unchanged. If the null value is sent, the old value should be changed to null. (For further details, see Section 2.11, "Message construction rules," - step 2d.)

The various chapters of the Standard contain segment attribute tables. These tables list and describe the data fields in the segment and characteristics of their usage. A comprehensive data dictionary of all HL7 fields is provided in Appendix A. In defining a segment, the following information is specified about each field:

2.7.1 Position (sequence within the segment)

Definition: Ordinal position of the data field within the segment. This number is used to refer to the data field in the text comments that follow the segment definition table.

In the segment attribute tables this information is provided in the column labeled **SEQ**.

2.7.2 Maximum length

Definition: Maximum number of characters that one occurrence of the data field may occupy.

In the segment attribute tables this information is in a column labeled **LEN.**

The maximum length is not of conceptual importance in the abstract message or the HL7 coding rules. The length of a field is normative, but can be changed on a site specific basis. It is calculated to include the component and subcomponent separators that are defined below. Because the maximum length is that of a single occurrence, the repetition separator is not included in calculating the maximum length (See Section 2.7.5, “Repetition”). A composite data type may not have a maximum length less than the maximum length of its largest component data type (i.e., in PID-3, CX includes HD, which in turn includes an IS, ID, and ST).

The following conventions have been applied:

1) The maximum length of the data field shall be expressed as a number.
2) If the maximum length needs to convey the notion of a Very Large Number, the number 65536 should be displayed to alert the user. This convention takes the place of the practice in versions prior to 2.4 of abbreviating this expression as 64K.
3) If the maximum length cannot be definitively expressed because the data type for the field is variable, the symbolic number 99999 should be displayed. This convention takes the place of the practice in versions prior to 2.4 of displaying the notation “varies” or some other non-numeric description.

The following maximum field lengths are specified:

Maximum Field Lengths

Field Type	Data Type	Length
Coded fields:	CE	250
	CX	250
	CNE	250
	CWE	250
	CK	250
	CN	250
Phone number field:	XTN	250

Name fields:	XCN	250
	XPN	250
	XON	250
	PPN	250
Address fields:	XAD	250

2.7.3 Data type

Definition: Restrictions on the contents of the data field.

In the segment attribute tables this information is provided in the column labeled **DT**. If the data type of the field is variable, the notation "varies" will be displayed.

There are a number of data types defined by HL7. These are explained in Section 2.9, "Data types."

2.7.4 Optionality

Definition: Whether the field is required, optional, or conditional in a segment.

In the segment attribute tables this information is provided in the column labeled **OPT**.

The designations for optionality are:

R - required

O - optional

C - conditional on the trigger event or on some other field(s). The field definitions following the segment attribute table should specify the algorithm that defines the conditionality for this field.

X - not used with this trigger event

B - left in for backward compatibility with previous versions of HL7. The field definitions following the segment attribute table should denote the optionality of the field for prior versions.

Note: For Versions 2.3 and higher: the optionality of fields should be explicitly documented in the segment field definitions that follow each segment definition table; if the optionality of fields within a segment changes depending on the trigger event, that optionality should also be explicitly documented.

For fields defined by HL7 data types containing multiple components or subcomponents, the optionality of a given component or subcomponent must be specified in the detailed field definitions that follow the formal segment attribute tables. (See also Sections 2.8, "MESSAGE DELIMITERS," 2.9, "Data types," and 2.11, "Message construction rules").

2.7.5 Repetition

Definition: Whether the field may repeat.

In the segment attribute tables this information is provided in the column labeled **RP/#.**

The designations for Repetition are:

N or blank	-	no repetition
Y	-	the field may repeat an indefinite or site-determined number of times
(integer)	-	the field may repeat up to the number of times specified by the integer

Each occurrence may contain the number of characters specified by the field's maximum length. (See Section 2.7.2, "Maximum length.")

Usage Note: For improved readability some technical committees opt to leave the Repetition fields blank to indicate that the field may NOT repeat. A blank may NOT be construed to mean that the field may optionally repeat.

2.7.6 Table

Definition: The table attribute of the data field definition specifies the HL7 identifier for a set of coded values.

In the segment attribute tables, the table identifier is provided in the column labeled **TBL#.** An entry in the table number column means that the table name and the element name are equivalent. If this attribute is not valued or blank, there is not a table of values defined for the field.

A number of conventions have been applied to this attribute of the data field definition.

1. If more than one table is applicable, the format xxxx/yyyy will be used to so designate multiple tables. Details on multiple tables will be specified in field notes.

2. If the field is of data type ID or IS a table number will be allocated even if, in the case of IS, there may be a notation "No Suggested values"

3. If the field is of data type CE and one or more externally or locally defined tables may be used, the symbolic number 9999 will appear in the column. This is to indicate that table values are used, but no HL7/User Defined table can be allocated. The narrative may constrain which external tables can be used.

4. Tables embedded in field components or subcomponents will not be cited in the attribute column. The exception to this convention are the CE, CNE, CWE and CF data types where the table is dependent on the field context. This also applies to the CM data type if it contains embedded tables.

   a) Data types having embedded tables are identified in *HL7 Table 440 -Data types*. These tables are defined in the data type section. They may, however, be constrained in the field note section. The field note definition supercedes the definition in the data type section.

   b) Tables embedded in fields with a data type of CE, CF CNE, CM or CWE are only defined in the field notes section.

HL7 defines table values in 3 ways: HL7 defined, user-defined and externally defined

User-defined Tables: A user-defined table is a set of values that are locally or site defined. This accommodates certain fields, like *PV1-3 - Assigned patient location,* that will have values that vary from institution to institution. Even though these tables are not defined in the Standard, they are given a user-defined table number to facilitate implementations. HL7 sometimes publishes suggested values that a site may use

as a starter set (e.g., *table 0001- Sex*). The IS data type is often used to encode values for these tables. Note that some of these tables (e.g., *table - 0302 Point of care*) may reference common master files.

There are some user-defined tables that contain values that might be standardized across institutions but for which no applicable official standard exists. For these a set of **suggested** values may be listed in Appendix A. These suggested values appear in the text in a standard box format (e.g., *HL7 Table 0062 - Event reason* in Section 3.4.1.4, "Event reason code"). It is recommended that these values be used where applicable within an institution and serve as a basis for extensions as required. These values may, however, be redefined locally. The appropriate functional committee within HL7 solicits suggestions for additional values from institutions that are applying the Standard.

HL7 Tables: An HL7 table is a set of values defined and published by HL7. They are a part of the HL7 Standard because they affect the interpretation of the messages that contain them. These values may not be redefined locally; however, the table itself may be extended to accommodate locally defined values. This is particularly applicable in the case of *HL7 table 0003 – Event Type*. The ID data type is most often used to encode values for HL7 tables. The values are listed in Appendix A. These HL7 tables also appear in the text in a standard box format (e.g., *HL7 table 0003 Event Type*)

External Tables: An external table is a set of coded values defined and published by another standards organization. External tables are used to populate fields like *FT1-19-Diagnosis Code - FT1*. Another example, the encoding of clinical observations using LOINC codes. The CE data type is used to represent values for these fields.

Table numbers 9000 and above are reserved for externally-defined tables published by HL7. Such tables arise from applications where the concepts and possibly the codes are established by external agencies due to regulatory requirements or agreements between HL7 and other Standards Developing Organizations. They are published by HL7 on behalf of other organizations. Their contents are not subject to approval by HL7 ballot. Such tables will be published with HL7 Standards. However, they may be updated more frequently than HL7 Standards. HL7 will provide free downloads of the most recent versions of these tables via the Internet without requiring membership in HL7.

2.7.7 ID number

Small integer that uniquely identifies the data item throughout the Standard. In the segment definition this information is provided in the column labeled **ITEM #.**

2.7.8 Name

Descriptive name for the data item. In the segment attribute tables this information is provided in the column labeled **ELEMENT NAME**.

When the same name is used in more than one segment, it must have the same data type and semantic meaning in each segment as well as the same ID number. To deal with any ambiguities arising from this convention, whenever a field is referenced herein, the segment name and position must always be included.

2.8 MESSAGE DELIMITERS

In constructing a message, certain special characters are used. They are the segment terminator, the field separator, the component separator, subcomponent separator, repetition separator, and escape character. The segment terminator is always a carriage return (in ASCII, a hex 0D). The other delimiters are defined in the MSH segment, with the field delimiter in the 4th character position, and the other delimiters occurring as in the field called Encoding Characters, which is the first field after the segment ID. The delimiter values used in the MSH segment

are the delimiter values used throughout the entire message. In the absence of other considerations, HL7 recommends the suggested values found in Figure 2-1 delimiter values.

At any given site, the subset of the possible delimiters may be limited by negotiations between applications. This implies that the receiving applications will use the agreed upon delimiters, as they appear in the Message Header segment (MSH), to parse the message.

Figure 2-1. Delimiter values

Delimiter	Suggested Value	Encoding Character Position	Usage
Segment Terminator	<cr> (hex 0D)	-	Terminates a segment record. This value cannot be changed by implementors.
Field Separator	|	-	Separates two adjacent data fields within a segment. It also separates the segment ID from the first data field in each segment.
Component Separator	^	1	Separates adjacent components of data fields where allowed.
Subcomponent Separator	&	4	Separates adjacent subcomponents of data fields where allowed. If there are no subcomponents, this character may be omitted.
Repetition Separator	~	2	Separates multiple occurrences of a field where allowed.
Escape Character	\	3	Escape character for use with any field represented by an ST, TX or FT data type, or for use with the data (fourth) component of the ED data type. If no escape characters are used in a message, this character may be omitted. However, it must be present if subcomponents are used in the message.

2.9 DATA TYPES

The data types in this section are listed in alphabetical order.

Note: For data types which contain multiple components or subcomponents, the examples given in this section do not specify the optionality of the component or subcomponents. This must be specified in the field definitions that follow the formal segment attribute tables.

Except for the TS data type and the maximum or minimum lengths for several other data types (CE, PN, TX, FT), the field length of HL7 attributes is specified in the segment attribute tables, and any specific length of the components or subcomponents of those attributes must be specified in the field definitions that follow the formal segment attribute tables. In general, HL7 does not specify the lengths of components and/or subcomponents.

(The data type examples in this Standard are given using the standard HL7 encoding rules, with the delimiter values from Figure 2-1 of Section 2.8, "MESSAGE DELIMITERS." Although only one set of encoding rules is defined as a standard in HL7 Version 2.4, other encoding rules are possible (but since they are non-standard, they may only be used by a site-specific agreement).

In certain data type definitions, square brackets, "[" and "]", are used to specify optional parts of a data type (or of a data type component or subcomponent).

The following table lists the data types by category. An alpha listing of the data types is also available. See *HL7 Table 0440*.

Figure 2-2. HL7 data types by category

Data Type Category/ Data type	Data Type Name	LEN	HL7 Section Reference	Notes/Format
Alphanumeric				
ST	String	199	2.9.43	
TX	Text data	65536	2.9.48	
FT	Formatted text	65536	2.9.20	
SRT	Sort order		2.9.42	<sort-by field/parameter (varies)> ^ <sequencing (ID)>
Numerical				
CQ	Composite quantity with units		2.9.10	<quantity (NM)> ^ <units (CE)>
MO	Money		2.9.26	<quantity (NM)> ^ <denomination (ID)>
NM	Numeric		2.9.28	
SI	Sequence ID		2.9.40	
SN	Structured numeric		2.9.41	<comparator (ST)> ^ <num1 (NM)> ^ <separator/suffix (ST)> ^ <num2 (NM)>
Identifier				
ID	Coded values for HL7 tables		2.9.22	
IS	Coded value for user-defined tables		2.9.23	
VID	Version identifier		2.9.50	<version ID (ID)> ^ <internationalization code (CE)> ^ <international version ID (CE)
HD	Hierarchic designator		2.9.21	<namespace ID (IS)> ^ <universal ID (ST)> ^ <universal ID type (ID)>
EI	Entity identifier		2.9.17	<entity identifier (ST)> ^ <namespace ID (IS)> ^ <universal ID (ST)> ^ <universal ID type (ID)>
RP	Reference pointer		2.9.37	<pointer (ST) > ^ < application ID (HD)> ^ <type of data (ID)> ^ <subtype (ID)>
PL	Person location		2.9.29	<point of care (IS)> ^ <room (IS)> ^ <bed (IS)> ^ <facility (HD)> ^ < location status (IS)> ^ <person location type (IS)> ^ <building (IS)> ^ <floor (IS)> ^ <location description (ST)>
PT	Processing type		2.9.32	<processing ID (ID)> ^ <processing mode (ID)>
Date/Time				
DT	Date		2.9.15	YYYY[MM[DD]]
TM	Time		2.9.44	HH[MM[SS[.S[S[S[S]]]]]][+/-ZZZZ]
TS	Time stamp		2.9.47	YYYY[MM[DD[HHMM[SS[.S[S[S[S]]]]]]][+/-ZZZZ] ^ <degree of precision>

Data Type Category/ Data type	Data Type Name	LEN	HL7 Section Reference	Notes/Format
Code Values				
CE	Coded element	250	2.9.3	<identifier (ST)> ^ <text (ST)> ^ <name of coding system (IS)> ^ <alternate identifier (ST)> ^ <alternate text (ST)> ^ <name of alternate coding system (IS)>
CNE	Coded with no exceptions	250	2.9.8	<identifier (ST)> ^ <text (ST)> ^ <name of coding system (IS)> ^ <alternate identifier (ST)> ^ <alternate text (ST)> ^ <name of alternate coding system (IS)> ^ <coding system version ID (ST)> ^ alternate coding system version ID (ST)> ^ <original text (ST) >
CWE	Coded with exceptions	250	2.9.11	<identifier (ST)> ^ <text (ST)> ^ <name of coding system (IS)> ^ <alternate identifier (ST)> ^ <alternate text (ST)> ^ <name of alternate coding system (IS)> ^ <coding system version ID (ST)> ^ alternate coding system version ID (ST)> ^ <original text (ST) >
CF	Coded element with formatted values		2.9.4	<identifier (ID)> ^ <formatted text (FT)> ^ <name of coding system (IS)> ^ <alternate identifier (ID)> ^ <alternate formatted text (FT)> ^ <name of alternate coding system (IS)>
CK	Composite ID with check digit	250	2.9.5	<ID number (NM)> ^ <check digit (NM)> ^ <code identifying the check digit scheme employed (ID)> ^ < assigning authority (HD)>
CN	Composite ID number and name	250	2.9.7	<ID number (ST)> ^ <family name (FN)> ^ <given name (ST)> ^ < second and further given names or initials thereof (ST)> ^ <suffix (e.g., JR or III) (ST)> ^ <prefix (e.g., DR) (ST)> ^ <degree (e.g., MD) (IS)> ^ <source table (IS)> ^ <assigning authority (HD)>
CX	Extended composite ID with check digit	250	2.9.12	<ID (ST)> ^ <check digit (ST)> ^ <code identifying the check digit scheme employed (ID)> ^ < assigning authority (HD)> ^ <identifier type code (ID)> ^ < assigning facility (HD) ^ <effective date (DT)> ^ <expiration date (DT)>

Data Type Category/ Data type	Data Type Name	LEN	HL7 Section Reference	Notes/Format
XCN	Extended composite ID number and name	250	2.9.52	In Version 2.3 and later, use instead of the CN data type. <ID number (ST)> ^ <family name (FN)> ^ <given name (ST)> ^ <second and further given names or initials thereof (ST)> ^ <suffix (e.g., JR or III) (ST)> ^ <prefix (e.g., DR) (ST)> ^ <degree (e.g., MD) (IS)> ^ <source table (IS)> ^ <assigning authority (HD)> ^ <name type code (ID)> ^ <identifier check digit (ST)> ^ <code identifying the check digit scheme employed (ID)> ^ <identifier type code (IS)> ^ <assigning facility (HD)> ^ <name representation code (ID)> ^ <name context (CE)> ^ <name validity range (DR)> ^ < name assembly order (ID)>
Generic				
CM	Composite		2.9.6	No new CM's are allowed after HL7 Version 2.2. ***The CM data type is maintained strictly for backward compatibility and may not be used for the definition of new fields.***
Demographics				
AD	Address		2.9.1	<street address (ST)> ^ < other designation (ST)> ^ <city (ST)> ^ <state or province (ST)> ^ <zip or postal code (ST)> ^ <country (ID)> ^ <address type (ID)> ^ <other geographic designation (ST)>
FN	Family name		2.9.19	<surname (ST)> ^ <own surname prefix (ST)> ^ <own surname (ST)> ^ <surname prefix from partner/spouse (ST)> ^ <surname from partner/spouse (ST)> Note: Appears ONLY in the PN and other PN-containing data types (PPN, XCN, XPN).
PN	Person name		2.9.30	<family name (FN)>^ <given name (ST) ^ < second and further given names or initials thereof (ST)> ^ <suffix (e.g., JR or III) (ST)> ^ <prefix (e.g., DR) (ST)> ^ <degree (e.g., MD) (IS)>
SAD	Street Address		2.9.38	<street or mailing address (ST)> ^ <street name (ST)> ^ <dwelling number (ST)> Note: Appears ONLY in the XAD data type.
TN	Telephone number		2.9.45	[NN] [(999)]999-9999[X99999][B99999][C any text]
XAD	Extended address	250	2.9.51	In Version 2.3 and later, replaces the AD data type. <street address (SAD)> ^ <other designation (ST)> ^ <city (ST)> ^ <state or province (ST)> ^ <zip or postal code (ST)> ^ <country (ID)> ^ < address type (ID)> ^ <other geographic designation (ST)> ^ <county/parish code (IS)> ^ <census tract

Data Type Category/ Data type	Data Type Name	LEN	HL7 Section Reference	Notes/Format
				(IS)> ^ <address representation code (ID)> ^ <address validity range (DR)>
XPN	Extended person name	250	2.9.54	In Version 2.3, replaces the PN data type. <family name (FN)> ^ <given name (ST)> ^ <second and further given names or initials thereof (ST)> ^ <suffix (e.g., JR or III) (ST)> ^ <prefix (e.g., DR) (ST)> ^ <degree (e.g., MD) (IS)> ^ <name type code (ID) > ^ <name representation code (ID)> ^ <name context (CE)> ^ <name validity range (DR)> ^ <name assembly order (ID)>
XON	Extended composite name and ID number for organizations	250	2.9.53	<organization name (ST)> ^ <organization name type code (IS)> ^ <ID number (NM)> ^ <check digit (NM)> ^ <code identifying the check digit scheme employed (ID)> ^ <assigning authority (HD)> ^ <identifier type code (IS)> ^ <assigning facility ID (HD)> ^ <name representation code (ID)>
XTN	Extended telecommunications number	250	2.9.55	In Version 2.3 and later, replaces the TN data type. [**N**NN] [(999)]999-9999 [X99999] [B99999] [C any text] ^ <telecommunication use code (ID)> ^ <telecommunication equipment type (ID)> ^ <email address (ST)> ^ <country code (NM)> ^ <area/city code (NM)> ^ <phone number (NM)> ^ <extension (NM)> ^ <any text (ST)>
Specialty/Chapter Specific				
Waveform				
CD	Channel definition		2.9.2	For waveform data only, see Chapter 7, Section 7.16.2. <channel identifier (CM)> ^ <waveform source (CM)> ^ <channel sensitivity/units (CM) > ^ <channel calibration parameters (CM)> ^ <sampling frequency (NM)> ^ <minimum/maximum data values (CM)>
MA	Multiplexed array		2.9.25	For waveform data only, see Chapter 7, Section 7.15.2. <sample 1 from channel 1 (NM)> ^ <sample 1 from channel 2 (NM)> ^ <sample 1 from channel 3 (NM)> ...~<sample 2 from channel 1 (NM)> ^ <sample 2 from channel 2 (NM)> ^ <sample 2 from channel 3 (NM)> ...~
NA	Numeric array		2.9.27	For waveform data only, see Chapter 7, Section 7.15.1. <value1 (NM)> ^ <value2 (NM)> ^ <value3 (NM)> ^ <value4 (NM)> ^ ...
ED	Encapsulated data		2.9.16	Supports ASCII MIME-encoding of binary data. <source application (HD) > ^ <type of data (ID)> ^ <data subtype (ID)> ^ <encoding (ID)> ^ <data (ST)>

Data Type Category/ Data type	Data Type Name	LEN	HL7 Section Reference	Notes/Format
Price Data				
CP	Composite price		2.9.9	In Version 2.3, replaces the MO data type. <price (MO)> ^ <price type (ID)> ^ <from value (NM)> ^ <to value (NM)> ^ <range units (CE)> ^ <range type (ID)>
Patient Administration /Financial Information				
FC	Financial class		2.9.18	<financial class (IS)> ^ <effective date (TS)>
Extended Queries				
QSC	Query selection criteria		2.9.34	<segment field name (ST)> ^ <relational operator (ID)> ^ <value (ST)> ^ <relational conjunction (ID)>
QIP	Query input parameter list		2.9.33	<segment field name (ST) > ^ <value1 (ST) & value2 (ST) & value3 (ST) ...>
RCD	Row column definition		2.9.35	<segment field name (ST)> ^ <HL7 data type (ID)> ^ <maximum column width (NM)>
Master Files				
DLN	Driver's license number		2.9.13	<license number (ST)> ^ <issuing state, province, country (IS)> ^ <expiration date (DT)>
JCC	Job code/class		2.9.24	<job code (IS)> ^ <job class (IS)>
VH	Visiting hours		2.9.49	<start day range (ID)> ^ <end day range (ID)> ^ <start hour range (TM)> ^ <end hour range (TM)>
Medical Rec-ords/Information Management				
PPN	Performing person time stamp	250	2.9.31	<ID number (ST)> ^ <family name (FN)> ^ <given name (ST) ^ <second and further given names or initials thereof (ST)>^ <suffix (e.g., JR or III) (ST)> ^ <prefix (e.g., DR) (ST)> ^ <degree (e.g., MD) (IS)> ^ <source table (IS)> ^ <assigning authority (HD)> ^ <name type code(ID)> ^ <identifier check digit (ST)> ^ <code identifying the check digit scheme employed (ID)> ^ <identifier type code (IS)> ^ <assigning facility (HD)> ^ < date/time action performed (TS)> ^ <name representation code (ID)> ^ <name context (CE)> ^ <name validity range (DR)> ^ <name assembly order (ID)>
Time Series:				
DR	Date/time range		2.9.14	<range start date/time (TS)> ^ <range end date/time (TS)>

Data Type Category/ Data type	Data Type Name	LEN	HL7 Section Reference	Notes/Format
RI	Repeat interval		2.9.36	Scheduling Chapter Only: <repeat pattern (IS)> ^ <explicit time interval (ST)>
SCV	Scheduling class value pair		2.9.39	Scheduling Chapter Only: <parameter class (IS)> ^ <parameter value (ST)>
TQ	Timing/quantity		2.9.46	For timing/quantity specifications for orders, see Chapter 4, Section 4.3. <quantity (CQ)> ^ <interval (*)> ^ <duration (*)> ^ <start date/time (TS)> ^ <end date/time (TS)> ^ <priority (ST)> ^ <condition (ST)> ^ <text (TX)> ^ <conjunction (ID)> ^ <order sequencing (*)> ^ <occurrence duration (CE)> ^ <total occurrences (NM)>

* for subcomponents of these elements please refer to the definition in the text.

2.9.1 AD - address

```
Components: <street address (ST)> ^ < other designation (ST)> ^ <city (ST)> ^ <state or province
            (ST)> ^ <zip or postal code (ST)> ^ <country (ID)> ^ <address type (ID)> ^ <other
            geographic designation (ST)>
```

Note: Replaced by the XAD data type as of v 2.3.

Example:

```
|10 ASH LN^#3^LIMA^OH^48132|
```

2.9.1.1 Street address (ST)

The street or mailing address of a person or institution. When referencing an institution, this first component is used to specify the institution name. When used in connection with a person, this component specifies the first line of the address.

2.9.1.2 Other designation (ST)

Second line of address. In general, it qualifies address. Examples: Suite 555 or Fourth Floor. When referencing an institution, this component specifies the street address.

2.9.1.3 City (ST)

2.9.1.4 State or province (ST)

State or province should be represented by the official postal service codes for that country.

2.9.1.5 Zip or postal code (ST)

Zip or postal codes should be represented by the official codes for that country. In the US, the zip code takes the form 99999[-9999], while the Canadian postal code takes the form A9A9A9.

2.9.1.6 Country (ID)

Defines the country of the address. ISO 3166 provides a list of country codes that may be used.[1] This ISO table has three separate forms of the country code: HL7 specifies that the 3-character (alphabetic) form be used for the country code. Refer to *HL7 Table 0399 – Country code* for valid values.

2.9.1.7 Address type (ID)

Type is optional and defined by *HL7 Table 0190 - Address type*.

HL7 Table 0190 - Address type

Value	Description
BA	Bad address
N	Birth (nee) (birth address, not otherwise specified)
BDL	Birth delivery location (address where birth occurred)
F	Country Of Origin
C	Current Or Temporary
B	Firm/Business
H	Home
L	Legal Address
M	Mailing
O	Office
P	Permanent
RH	Registry home. Refers to the information system, typically managed by a public health agency, that stores patient information such as immunization histories or cancer data, regardless of where the patient obtains services.
BR	Residence at birth (home address at time of birth)

2.9.1.8 Other geographic designation (ST)

Other geographic designation includes county, bioregion, SMSA, etc.

2.9.2 CD - channel definition

```
Components: <channel identifier (CM)> ^ <waveform source (CM)> ^ <channel sensitivity/units (CM)>
            ^ <channel calibration parameters (CM)> ^ <sampling frequency (NM)> ^ <mini-
            mum/maximum data values (CM)>
```

This data type is used for labeling of digital waveform data. See Chapter 7, Section 7.16.2, “CD - channel definition,” for a complete description of this data type.

2.9.3 CE - coded element

```
Components: <identifier (ST)> ^ <text (ST)> ^ <name of coding system (IS)> ^ <alternate identifier
            (ST)> ^ <alternate text (ST)> ^ <name of alternate coding system (IS)>
```

Length: 250

[1] Available from ISO 1 Rue de Varembe, Case Postale 56, CH 1211, Geneva, Switzerland.

This data type transmits codes and the text associated with the code.

Example:

```
|F-11380^CREATININE^I9^2148-5^CREATININE^LN|
```

2.9.3.1 Identifier (ST)

Sequence of characters (the code) that uniquely identifies the item being referenced by the <text>. Different coding schemes will have different elements here.

2.9.3.2 Text (ST)

Name or description of the item in question. E.g., myocardial infarction or X-ray impression. Its data type is string (ST).

2.9.3.3 Name of coding system (IS)

Each coding system is assigned a unique identifier. This component will serve to identify the coding scheme being used in the identifier component. The combination of the **identifier** and **name of coding system** components will be a unique code for a data item. Each system has a unique identifier.

User-defined Table 0396 – Coding system contains the allowable values. The table includes ASTM E1238-94, Diagnostic, procedure, observation, drug ID, and health outcomes coding systems as identified in the tables in Section 7.1.4, "Coding schemes." Others may be added as needed.

Some organizations that publish code sets author more than one. The coding system, then, to be unique is a concatenation of the name of the coding authority organization and the name of its code set or table. When an HL7 table is used for a CE data type, the ***name of coding system*** component is defined as ***HL7nnnn*** where ***nnnn*** is the HL7 table number. Similarly, ISO tables will be named ISOnnnn, where nnnn is the ISO table number.

2.9.3.4 Alternate identifier (ST)

For explanation, see text after 2.8.3.6.

2.9.3.5 Alternate text (ST)

For explanation, see text after 2.8.3.6.

2.9.3.6 Name of alternate coding system (IS)

Note on the Alternate components (4, 5, 6) (for components 1, 2, 3)

These three components are defined analogously to the above for the alternate or local coding system. If the *alternate text* component is absent, and the alternate identifier is present, the *alternate text* will be taken to be the same as the *text* component. If the *alternate coding system* component is absent, it will be taken to mean the locally-defined system.

Note:	The presence of two sets of equivalent codes in this data type is semantically different from a repetition of a CE-type field. With repetition, several distinct codes (with distinct meanings) may be transmitted.

Refer to *User-defined table 0396 Coding Systems* for valid values. When an HL7 table is used for a CE data type, the ***name of coding system*** component is defined as ***HL7nnnn*** where ***nnnn*** is the HL7 table number. Guidelines for the diagnostic, procedure, observation, drug, and health outcomes coding systems use are presented in Chapter 7.

2.9.4 CF - coded element with formatted values

This data type transmits codes and the formatted text associated with the code. This data type can be used to transmit for the first time the formatted text for the **canned text** portion of a report, for example, a standard radiologic description for a normal chest X-ray. The receiving system can store this information and in subsequent messages only the identifier need be sent. Another potential use of this data type is transmitting master file records that contain formatted text. This data type has six components as follows:

```
Components: <identifier (ID)> ^ <formatted text (FT)> ^ <name of coding system (IS)> ^ <alternate
            identifier (ID)> ^ <alternate formatted text (FT)> ^ <name of alternate coding system
            (IS)>
```

The components, primary and alternate, are defined exactly as in the CE data type with the exception of the second and fifth components, which are of the formatted text data type. Example:

```
OBX||CF|71020^CXR^CPMC||79989^\H\Description:\N\\.sp\\ti+4\Heart is not enlarged.
    There is no evidence of pneumonia, effusion, pneumothorax or any masses.
    \.sp+3\\H\Impression:\N\\.sp\\.ti+4\Negative chest.^CPMC
```

2.9.5 CK - composite ID with check digit

```
Components:  <ID number (NM)> ^ <check digit (NM)> ^ <code identifying the check digit scheme em-
             ployed (ID)> ^ < assigning authority (HD)>
```

Length: 250

This data type is used only in *CDM-11-Contract number* as defined in chapter 8, section 8.10.2.11. If a site is not using check digits for a particular CK field, the second and third components are not valued. Example:

```
|128952^6^M11^ADT01|
```

2.9.5.1 ID number (NM)

2.9.5.2 Check digit (NM)

The check digit in this data type is not an add-on produced by the message processor. It is the check digit that is part of the identifying number used in the sending application. If the sending application does not include a self-generated check digit in the identifying number, this component should be valued null.

2.9.5.3 Code identifying the check digit scheme employed (ID)

The check digit scheme codes are defined in *HL7 Table 0061 - Check digit scheme*.

HL7 Table 0061 - Check digit scheme

Value	Description
NPI	Check digit algorithm in the US National Provider Identifier
ISO	ISO 7064: 1983
M10	Mod 10 algorithm
M11	Mod 11 algorithm

The algorithm for calculating a Mod10 check digit is as follows:

Assume you have an identifier = 12345. Take the odd digit positions, counting from the right, i.e., 531, multiply this number by 2 to get 1062. Take the even digit positions, starting from the right (i.e., 42), prepend these to the 1062 to get 421062. Add all of these six digits together to get 15. Subtract this number

from the next highest multiple of 10, i.e., 20 - 15 to get 5. The Mod10 check digit is 5. The Mod10 check digit for 401 is 0; for 9999, it's 4; for 99999999, it's 8.

The algorithm for calculating a Mod11 check digit is as follows:

Terms

d	=	digit of number starting from units digit, followed by 10's position, followed by 100's position, etc.
w	=	weight of digit position starting with the units position, followed by 10's position, followed by 100's position etc. Values for w = 2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 7, etc. (repeats for each group of 6 digits)
c	=	check digit

Calculation

(Step 1) m	=	sum of (d * w) for positions 1, 2, etc. starting with units digit for d = digit value starting with units position to highest order for w = weight value from 2 to 7 for every six positions starting with units digit
(Step 2) c1	=	m mod 11
(Step 3) if c1	=	0 then reset c1 = 1
(Step 4)	=	(11 - c1) mod 10

Example:

```
if the number is 1234567, then the mod 11 check digit = 4
```

The calculations are:

```
M  = (7*2)+(6*3)+(5*4)+(4*5)+(3*6)+(2*7)+(1*2)
   = 14 + 18 + 20 + 20 + 18 + 14 + 2
   = 106
c1 = 106 mod 11
   = 7
c  = (11-c1) mod 10
   = 4 mod 10
   = 4
```

Other variants of these check digit algorithms exist and may be used by local bilateral site agreement.

2.9.5.4 Assigning authority (HD)

The assigning authority is a unique identifier of the system (or organization or agency or department) that creates the data. It is a HD data type. Assigning authorities are unique across a given HL7 implementation. *User-defined Table 0363 – Assigning authority* is used as the HL7 identifier for the user-defined table of values for the first sub-component, namespace ID.

User-defined Table 0363 – Assigning authority

Value	Description
AUSDVA	Australia - Dept. of Veterans Affairs
AUSHIC	Australia - Health Insurance Commission
CANAB	Canada - Alberta
CANBC	Canada - British Columbia
CANMB	Canada - Manitoba
CANNB	Canada - New Brunswick
CANNF	Canada - Newfoundland
CANNS	Canada - Nova Scotia
CANNT	Canada - Northwest Territories
CANNU	Canada - Nanavut
CANON	Canada - Ontario
CANPE	Canada - Prince Edward Island
CANQC	Canada - Quebec
CANSK	Canada - Saskatchewan
CANYT	Canada - Yukon Territories
NLVWS	NL - Ministerie van Volksgezondheid, Welzijn en Sport
USCDC	US Center for Disease Control
USHCFA	US Healthcare Finance Authority
USSSA	US Social Security Administration

Note: When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID* (referenced by the first sub-component of the HD component) may be redefined (given a different user-defined table number and name) by the technical committee responsible for that segment.

By site agreement, implementors may continue to use *User-defined Table 0300 – Namespace ID* for the first sub-component.

2.9.6 CM - composite

A field that is a combination of other meaningful data fields. Each portion is called a **component**. The specific components of CM fields are defined within the field descriptions. Certain other composites have been separately identified and are described below.

No new CMs are allowed after HL7 version 2.2. ***The CM data type is maintained strictly for backward compatibility and may not be used for the definition of new fields.***

Wherever a component of an HL7 field is itself an HL7 data type which contains components, its delimiters are demoted by one. Thus a component designated as a CE data type should be encoded as `<identifier & text & name of coding system>` (see Section 2.9.3, “CE - coded element”). Note that since HL7 delimiters are not recursive, an HL7 data type containing components cannot be a subcomponent. When this level of detail is needed, each component of the HL7 data type can be encoded as a separate subcom-

ponent. For an example of this, see the encoding of the filler order number in the order sequencing component of the Timing/Quantity data type.

2.9.7 CN - composite ID number and name

```
Components: <ID number (ST)> ^ <family name (FN)> ^ <given name (ST)> ^ < second and further given
            names or initials thereof (ST)> ^ <suffix (e.g., JR or III) (ST)> ^ <prefix (e.g.,
            DR) (ST)> ^ <degree (e.g., MD) (IS)> ^ <source table (IS)> ^ <assigning authority
            (HD)>

Subcomponents of family name:  <surname (ST)> ^ <own surname prefix (ST)> ^ <own surname (ST)> ^
            <surname prefix from partner/spouse (ST)> ^ <surname from partner/spouse (ST)>
```

Note: Replaced by XCN data type as of v. 2.3.

Length: 250

This data type is used when identifying a person both as a coded value and with a text name. For specific fields, individual sites may elect to omit the ID or the name. Example:

|12372^RIGGINS^JOHN^""^""^""^MD^ADT1|

|12372^^^^^^^ADT1|

|^RIGGINS^JOHN^""^""^""^MD|

2.9.7.1 ID number (ST)

Coded ID according to a user-defined table, defined by the 8th component. If the first component is present, either the source table or the assigning authority must be valued.

2.9.7.2 Family name (FN)

This component allows full specification of the surname of a person. Where appropriate, it differentiates the person's own surname from that of the person's partner or spouse, in cases where the person's name may contain elements from either name. It also permits messages to distinguish the surname prefix (such as "van" or "de") from the surname root. See section 2.9.19, “FN - family name”.

2.9.7.3 Given name (ST)

First name.

2.9.7.4 second and further given names or initials thereof (ST)

2.9.7.5 Suffix (ST)

Used to specify a name suffix (e.g., Jr. or III).

2.9.7.6 Prefix (ST)

Used to specify a name prefix (e.g., Dr.).

2.9.7.7 Degree (IS)

Used to specify an educational degree (e.g., MD). Refer to *User-defined Table 0360 – Degree* for suggested values.

2.9.7.8 Source table (IS)

User-defined Table 0297 - CN ID source is used as the HL7 identifier for the user-defined table of values for this component. Used to delineate the first component.

User-defined Table 0297 – CN ID source

Value	Description
	No suggested values defined

2.9.7.9 Assigning authority (HD)

The assigning authority is a unique identifier of the system (organization or agency or department) that creates the data. It is a HD data type. *User-defined Table 0363 – Assigning authority* is used as the HL7 identifier for the user-defined table of values for the first sub-component of the HD data type, *namespace ID*.

Note: When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID*, (referenced by the first sub-component of the HD component) may be redefined (given a different user-defined table number and name) by the technical committee responsible for that segment.

By site agreement, implementors may continue to use *User-defined Table 0300 – Namespace ID* for the first sub-component.

2.9.8 CNE – coded with no exceptions

```
Components: <identifier (ST)> ^ <text (ST)> ^ <name of coding system (IS)> ^ <alternate identifier
            (ST)> ^ <alternate text (ST)> ^ <name of alternate coding system (IS)> ^ <coding sys-
            tem version ID (ST)> ^ alternate coding system version ID (ST)> ^ <original text
            (ST)>
```

Length: 250

2.9.8.1 Identifier (ST)

Sequence of characters (the code) that uniquely identifies the item being referenced by the <text>. Different coding schemes will have different elements here.

2.9.8.2 Text (ST)

Name or description of the item in question. E.g., myocardial infarction or X-ray impression. Its data type is string (ST). This is the corresponding text assigned by the coding system to the identifier.

2.9.8.3 Name of coding system (IS)

Each coding system is assigned a unique identifier. This component will serve to identify the coding scheme being used in the identifier component. The combination of the **identifier** and **name of coding system** components will be a unique code for a data item. Each system has a unique identifier.

User-defined Table 0396 – Coding system contains the allowable values. The table includes ASTM E1238-94, Diagnostic, procedure, observation, drug ID, and health outcomes coding systems as identified in the tables in Section 7.2.5, “Coding schemes.” Others may be added as needed.

Some organizations that publish code sets author more than one. The coding system, then, to be unique is a concatenation of the name of the coding authority organization and the name of its code set or table. When an HL7 table is used for a CE data type, the ***name of coding system*** component is defined as ***HL7nnnn*** where ***nnnn*** is the HL7 table number. Similarly, ISO tables will be named ISOnnnn, where nnnn is the ISO table number.

2.9.8.4 Alternate identifier (ST)

Analogous to “Identifier” above. See 2.9.8.10, “Usage notes:” for further description.

2.9.8.5 Alternate text (ST)

Analogous to “Text” above. See 2.9.8.10, “Usage notes:” for further description.

2.9.8.6 Name of alternate coding system (IS)

Analogous to “Name of Coding System” above. See 2.9.8.10, “Usage notes:” for further description.

2.9.8.7 Coding system version ID (ST)

This is the version ID for the coding system identified by component 1-3. It belongs conceptually to components 1-3 and appears here only for reasons of backward compatibility.

2.9.8.8 Alternate coding system version ID (ST)

This is the version ID for the coding system identified by components 4-6. It belongs conceptually to the group of Alternate components (see note 2.9.3.6) and appears here only for reasons of backward compatibility.

2.9.8.9 Original text (ST)

The original text that was available to an automated process or a human before a specific code was assigned. This component is optional.

2.9.8.10 Usage notes:

Components 1-3 and 7: The *identifier* is required and must be a valid code. *Coding system* must either be present and have a value from the set of allowed coding systems or if not present it will be interpreted to have the same meaning as if it had been valued with the code meaning “HL7 coding system.” *User-defined Table 0396 – Coding system* contains the allowable values. If the coding system is any system other than “HL7 coding system,” *version ID* must be valued with an actual version ID. If the coding system is “HL7 coding system,” *version ID* may have an actual value or it may be absent. If *version ID* is absent, it will be interpreted to have the same value as the HL7 version number in the message header. Text description of code is optional but its use should be encouraged since it makes messages easier to review for accuracy, especially during interface testing and debugging.

Component 9: This is the original text that was available to an automated process or a human before a specific code was assigned. This component is optional.

Components 3-6 and 8: These components are optional. They are used to represent the local or user seen code as described. If present, components 3-6 and 8 obey the same rules of use and interpretation as described for components 1-3 and 7. If both are present, the identifiers in component 4 and component 1 should have exactly the same meaning, i.e., they should be exact synonyms.

CNE usage note: The CNE data type should be used when a required or mandatory coded field is needed.

User-defined Table 0396 – Coding system contains the allowable values. The table includes ASTM E1238-94, diagnostic, procedure, observation, drug and health outcomes coding systems. When an HL7 table is used for a CE data type, the ***name of coding system*** component is defined as ***HL7nnnn*** where ***nnnn*** is the HL7 table number. Guidelines for their use are presented in Chapter 7, Section 7.1, “Introduction and Overview.”

Examples:

1. If the Value Type field (sequence 2) of the OBX segment was defined to be of type CNE, and the desired *value type* was a number, the shortest representation of the *value type* field would be identical to the current ID field syntax:

   ```
   OBX|1|NM|718-7^Hemoglobin^LN||13.4|GM/DL|14-18|N||S|F<cr>
   ```

   A more verbose representation of the same OBX segment that included *text* would be:

   ```
   OBX|1|NM^Numeric|718-7^Hemoglobin^LN||13.4|GM/DL|14-18|N||S|F<cr>
   ```

   An even more verbose representation of the same OBX segment that included *text* and *coding system* would be:

   ```
   OBX|1|NM^Numeric^HL70125|718-7^Hemoglobin^LN||13.4|GM/DL|14-18|N||S|F<cr>
   ```

   To retain the information about the code used in the original system that created the data, alternative coding scheme data could be included:

   ```
   OBX|1|NM^Numeric^HL70125^NUM^Number^99LAB|718-7^Hemoglobin^LN||13.4|GM/DL|14-18|N||S|F<cr>
   ```

   If in addition to the above, one wanted to capture the version of vocabulary being used, and the HL7 version was "2.3.1", and the 99LAB coding scheme version was "1.1", the field would appear as:

   ```
   OBX|1|NM^Numeric^HL70125^NUM^Number^99LAB^2.3.1^1.1|718-7^Hemoglobin^LN||13.4|GM/DL|14-
   18|N||S|F<cr>
   ```

   Furthermore, if one wanted to include the "user seen" text of the value format, and the user had seen "Decimal" as the field type on a data entry screen, the field would appear as:

   ```
   OBX|1|NM^Numeric^HL70125^NUM^Number^99LAB^2.3.1^1.1^Decimal|718-7^Hemoglobin^LN||13.4|GM/DL|14-
   18|N||S|F<cr>
   ```

   Finally, a user could use the abbreviated form for the primary identifier, and use the long form for the alternative identifier.

   ```
   OBX|1|NM^^^NUM^Number^99LAB^^1.1^Decimal|718-7^Hemoglobin^LN||13.4|GM/DL|14-18|N||S|F<cr>
   ```

2. If the *value type* field had been defined as a CNE field, and if the desired *value type* was not in the value set, **a valid OBX instance could not be created.** For example, if a laboratory system had an internal value type of "Decimal Range", since there is no corresponding *value type* available in HL7 table 0125, no valid OBX instance could be created. The following instance would be incorrect. In all valid instances of CNE fields, the identifier field ***must*** have a valid value from the specified table.

   Incorrect (no valid identifier)

   ```
   OBX|1|^^^DR^Decimal Range^99LAB^^1.1^Decimal Range|718-7^Hemoglobin^LN||13.4|GM/DL|14-
   18|N||S|F<cr>
   ```

3. If the *coding scheme* is anything other than an HL7 table identifier, the coding scheme must be a valid scheme from the coding schemes specified in Chapter 7. For example, if the *Observation Identifier* field (sequence 3) of the OBX segment was typed as a CNE field, and LOINC version 1.0k was being used as the source of values for *Observation Identifier*, then the following OBX instance would be valid:

   ```
   OBX|1|NM|718-7^Hemoglobin^LN^^^^1.0k||13.4|GM/DL|14-18|N||S|F<cr>
   ```

   However, the following OBX instance would be incorrect, since the coding scheme designation "LOCAL" is not in the list of valid coding scheme identifiers, nor does it conform to the rules described in Chapter 7 for creating valid "local" coding scheme identifiers.

Incorrect (invalid coding scheme)

```
OBX|1|NM|9587-2^Hemoglobin^LOCAL^^^^1.0k||13.4|GM/DL|14-18|N||S|F<cr>
```

A valid OBX instance using a local coding scheme "99LAB" would be allowed, since "99LAB" conforms to the rules for identifying local coding schemes as described in Chapter 7. The valid OBX instance would be represented as follows:

```
OBX|1|NM|9587-2^Hemoglobin^99LAB^^^^6.5||13.4|GM/DL|14-18|N||S|F<cr>
```

Finally, if the coding scheme is anything other than an HL7 table identifier, a version number must be present. The following OBX instance is incorrect because it is missing a valid version number even though the coding scheme LN (LOINC) is valid:

Incorrect (missing version number)

```
OBX|1|NM|718-7^Hemoglobin^LN||13.4|GM/DL|14-18|N||S|F<cr>
```

2.9.9 CP - composite price

```
Components:  <price (MO)> ^ <price type (ID)> ^ <from value (NM)> ^ <to value (NM)> ^ <range units
             (CE)> ^ <range type (ID)>

Subcomponents of price:  <quantity (NM)> & <denomination (ID)>
```

Note:	This data type is often used to define a repeating field within a given segment.

Note:	Replaces MO as of v 2.3.

Example:

```
|100.00&USD^UP^0^9^min^P~50.00&USD^UP^10^59^min^P~10.00&USD^UP^60^999^P~50.00&USD^AP~
    200.00&USD^PF

~80.00&USD^DC|
```

2.9.9.1 Price (MO)

The only required component; usually containing a decimal point. Note that each component of the MO data type (Section 2.9.26, "MO - money") is a subcomponent here.

2.9.9.2 Price type (ID)

A coded value, data type ID. Refer to *HL7 Table 0205 - Price type* for valid values.

HL7 Table 0205 - Price type

Value	Description
AP	administrative price or handling fee
DC	direct unit cost
IC	indirect unit cost
PF	professional fee for performing provider
TF	technology fee for use of equipment
TP	total price
UP	unit price, may be based on length of procedure or service

2.9.9.3 From value (NM)

Each is a NM data type; together they specify the "range." The range can be defined as either time or quantity. For example, the range can indicate that the first 10 minutes of the procedure has one price. Another repetition of the data type can use the range to specify that the following 10 to 60 minutes of the procedure is charged at another price per; a final repetition can specify that the final 60 to N minutes of the procedure at a third price.

Note that, if the <`price type`> component is TP, both <`from value`> and <`to value`> may be null.

2.9.9.4 To value (NM)

See <from value> above.

2.9.9.5 Range units (CE)

```
Subcomponents of range units:  <identifier (ST)> & <text (ST)> & <name of coding system (IS)> &
            <alternate identifier (ST)> & <alternate text (ST)> & <name of alternate coding sys-
            tem (IS)>
```

A coded value, data type CE, defined by the standard table of units for either time or quantity (see for example, the tables in Section 7.1.4, "Coding schemes"). This describes the units associated with the range, e.g., seconds, minutes, hours, days, quantity (*i.e.*, count); it is required if <`from value`> and <`to value`> are present.

2.9.9.6 Range type (ID)

Refers to *HL7 Table 0298 - CP range type* for valid values.

HL7 Table 0298 - CP range type

Value	Description
P	Pro-rate. Apply this price to this interval, pro-rated by whatever portion of the interval has occurred/been consumed
F	Flat-rate. Apply the entire price to this interval, do not pro-rate the price if the full interval has not occurred/been consumed

2.9.10 CQ - composite quantity with units

```
Components:  <quantity (NM)> ^ <units (CE)>
```

Note:	In future versions, CQ fields should be avoided because the same data can usually be sent as two separate fields, one with the value and one with the units as a CE data type.

Examples:

```
|123.7^kg|     kilograms is an ISO unit
|150^lb&&ANSI+|   weight in pounds is a customary US unit defined within ANSI+.
```

2.9.10.1 Quantity (NM)

2.9.10.2 Units (CE)

The units in which the quantity is expressed. Field-by-field, default units may be defined within the specifications. When the observation is measured in the default units, the units need not be transmitted. If the measure is recorded in units different from the default, the measurement units must be transmitted as the second component. If the units are ISO+ units, then units should be recorded as lowercase abbreviations as

specified in Chapter 7. If the units are ANSI or local, the units and the source table must be recorded as specified in Chapter 7. But in these cases the component separator should be replaced by the subcomponent delimiter

```
Subcomponents for units: <identifier (ST)> & <text (ST)> & <name of coding system (IS)> & <alter-
            nate identifier (ST)> & <alternate text (ST)> & <name of alternate coding system
            (IS)>
```

2.9.11 CWE – coded with exceptions

```
Components: <identifier (ST)> ^ <text (ST)> ^ <name of coding system (IS)> ^ <alternate identifier
            (ST)> ^ <alternate text (ST)> ^ <name of alternate coding system (IS)> ^ <coding sys-
            tem version ID (ST)> ^ alternate coding system version ID (ST)> ^ <original text
            (ST)>
```

Length: 250

2.9.11.1 Identifier (ST)

Sequence of characters (the code) that uniquely identifies the item being referenced by the <text>. Different coding schemes will have different elements here.

2.9.11.2 Text (ST)

Name or description of the item in question. E.g., myocardial infarction or X-ray impression.

2.9.11.3 Name of coding system (IS)

Each coding system is assigned a unique identifier. This component will serve to identify the coding scheme being used in the identifier component. The combination of the **identifier** and **name of coding system** components will be a unique code for a data item. Each system has a unique identifier.

User-defined Table 0396 – Coding system contains the allowable values. The table includes ASTM E1238-94, Diagnostic, procedure, observation, drug ID, and health outcomes coding systems as identified in the tables in Section 7.1.4, “Coding schemes.” Others may be added as needed.

Some organizations that publish code sets author more than one. The coding system, then, to be unique is a concatenation of the name of the coding authority organization and the name of its code set or table. When an HL7 table is used for a CE data type, the ***name of coding system*** component is defined as ***HL7nnnn*** where ***nnnn*** is the HL7 table number. Similarly, ISO tables will be named ISOnnnn, where nnnn is the ISO table number.

2.9.11.4 Alternate identifier (ST)

Analogous to “Identifier” above. See 2.9.11.10, ”Usage notes:” for further description.

2.9.11.5 Alternate text (ST)

Analogous to “Text” above. See 2.9.11.10, ”Usage notes:” for further description.

2.9.11.6 Name of alternate coding system (IS)

Analogous to “Name of Coding System” above. See 2.9.11.10, ”Usage notes:” for further description.

2.9.11.7 Coding system version ID (ST)

This is the version ID for the coding system identified by components 1-3. It belongs conceptually to the group of component 1-3 and appears here only for reasons of backward compatibility.

2.9.11.8 Alternate coding system version ID (ST)

This is the version ID for the coding system identified by components 4-6. It belongs conceptually to the group of alternate components (see note 0, “Analogous to “Text” above. See 2.9.11.10, ”Usage notes:” for further description.

Name of alternate coding system (IS)”) and appears here only for reasons of backward compatibility.

2.9.11.9 Original text (ST)

The original text that was available to an automated process or a human before a specific code was assigned

2.9.11.10 Usage notes:

This is a field that is generally sent using a code, but where the code may be omitted in exceptional instances or by site agreement. Exceptional instances arise when the coding system being used does not have a code to describe the concept in the text.

Components 1-3 & 7 are used in one of three ways:

1) **Coded:** The identifier contains a valid code from a coding system. The coding system must either be present and have a value from the set of allowed coding systems, or if not present, it will be interpreted to have the same meaning as if it had been valued with the code meaning “HL7 coding system.” *User-defined Table 0396 – Coding system* contains the allowable values. The table includes ASTM E1238-94, Diagnostic, procedure, observation, drug ID, and health outcomes coding systems as identified in the tables in Section 7.1.4, “Coding schemes.” If the coding system is any system other than “HL7 coding system”, version ID must be valued with an actual version ID. If the coding system is “HL7 coding system,” version ID may have an actual value or it may be absent. If version ID is absent, it will be interpreted to have the same value as the HL7 version number in the message header. Text description is optional, but its use should be encouraged to aid in readability of the message during testing and debugging.

   Example 1a: OBX segment where the observation identifier is a LOINC code and the observation value is being sent as a CWE value, and the value is taken from SNOMED International.

   ```
   OBX|1|CWE|883-9^ABO Group^LN|1|F-D1250^Type O^SNM3^^^^3.4|||N||F<cr>
   ```

   Example 1b: OBX segment where the observation identifier is a LOINC code and the observation value is being sent as an CWE value, and the value is taken from a (currently hypothetical) HL7 table.

   ```
   OBX|1|CWE|883-9^ABO Group^LN|1|O^Type O^HL74875^^^^2.3.1|||N||F<cr>
   ```

2) **Uncoded:** Text is valued, the identifier has no value, and coding system and version ID follow the same rules as discussed for option 1.

   Example 2: OBX segment where the observation identifier is a LOINC code and the observation value is being sent as an CWE value, and the value is sent as text because the correct clinical value, “Wesnerian” was not found in the set of allowed values.

   ```
   OBX|1|CWE|883-9^ABO Group^LN|1|^Wesnerian^SNM3^^^^3.4|||A||F<cr>
   ```

3) **Data missing:** The name of the coding system is “HL7 CE Status,” version ID is either a real version, or if not present it has the same meaning as the version in the message header, and the identifier takes its value from one of the allowed CE field statuses. The codes for the allowed CE field statuses are shown below and will be maintained in a table as part of the HL7 vocabulary. Text description of code is optional.

   Example 3: OBX segment where the observation identifier is a LOINC code and the observation value is being sent as an LCE value, and no value can be sent because the test was not done.

```
OBX|1|CWE|883-9^ABO Group^LN|1|NAV^Not Available^HL70353^^^^2.3.1|||N||F<cr>
```

Component 9:

This is the original text that was available to an automated process or a human before a specific code was assigned. This field is optional.

Components 3-6 & 8:

Components 3-6 & 8 are optional. They are used to represent the local or user seen code. If present, components 3-6 & 8 obey the same rules of use and interpretation as described for components 1-3 & 7 (of the CWE data type). If both are present, the identifiers in component 4 and component 1 should have exactly the same meaning; i.e. they should be exact synonyms.

Example 4: OBX segment where the observation identifier is a LOINC code and the observation value is being sent as an CWE value, and the value is taken from SNOMED International. The user seen fields are being used to represent a local coding system (99LAB) used in the sending system.

```
OBX|1|CWE|883-9^ABO Group^LN|1|F-D1250^Type O^SNM3^O^O Type Blood^99LAB^3.4^|||||F<cr>
```

Summary of CWE usage notes with table of status values for various states without values:

The CWE data type should be used for coded fields that are optional or where it is permissible to send text for items that are not yet a part of the approved value set. In the normal situation, the identifier is valued with the code from the value set. If the value of the field is known, but is not part of the value set, then the value is sent as text, and the identifier has no value. If the field has an unknown status, then third form of the field is used (see **Data missing** above), and the appropriate status for the field is selected from the table of allowed statuses. When no code exists, use values from *HL7 Table 0353 – CWE statuses*

HL7 Table 0353 - CWE statuses

Code	Description
U	Unknown
UASK	Asked but Unknown
NAV	Not available
NA	Not applicable
NASK	Not asked

Where a text modifier might accompany a code, the “field” in the HL7 message would be of data type CWE and would be allowed to repeat. The first instance of the field would be used, as per option 1; i.e. the identifier would have a valid code. The second instance of the repeating field would be used, as per option 2, that is, the text description would take the value of the free text modifier.

2.9.12 CX - extended composite ID with check digit

```
Components: <ID (ST)> ^ <check digit (ST)> ^ <code identifying the check digit scheme employed
            (ID)> ^ < assigning authority (HD)> ^ <identifier type code (ID)> ^ < assigning fa-
            cility (HD) ^ <effective date (DT)> ^ <expiration date (DT)>
```

Length: 250

Example:

```
|1234567^4^M11^ADT01^MR^University Hospital|
```

This data type is used for specifying an identifier with its associated administrative detail.

2.9.12.1 ID (ST)

Definition: The value of the identifier itself. It is similar to the CK data type (see Section 2.9.5, "CK - composite ID with check digit") except that a ST data type is used instead of a NM data type.

2.9.12.2 Check digit (ST)

Defined as in the CK data type (see Section 2.9.5, "CK - composite ID with check digit") except that an ST data type is allowed instead of an NM data type. The check digit in this data type is not an add-on produced by the message processor. It is the check digit that is part of the identifying number used in the sending application. If the sending application does not include a self-generated check digit in the identifying number, this component should be valued null.

2.9.12.3 Code identifying the check digit scheme employed (ID)

Defined as in the CK data type (see Section 2.9.5, "CK - composite ID with check digit"). Refer to *HL7 Table 0061- Check digit scheme* for valid values.

Note:	The check digit and code identifying check digit scheme are null if ID is alphanumeric.

2.9.12.4 Assigning authority (HD)

The assigning authority is a unique name of the system (or organization or agency or department) that creates the data. It is a HD data type. *User-defined Table 0363 – Assigning authority* is used as the HL7 identifier for the user-defined table of values for the first sub-component of the HD component, <namespace ID>.

Note:	When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID* (referenced by the first sub-component of the HD component) may be re-defined (given a different user-defined table number and name) by the technical committee responsible for that segment. By site agreement, implementors may continue to use *User-defined Table 0300 – Namespace ID* for the first sub-component.

2.9.12.5 Identifier type code (ID)

A code corresponding to the type of identifier. In some cases, this code may be used as a qualifier to the "Assigning authority" component. Refer to *HL7 Table 0203 - Identifier type* for suggested values.

HL7 Table 0203 - Identifier type

Value	Description
AM	American Express
AN	Account number
BA	Bank Account Number
BR	Birth registry number
BRN	Breed Registry Number
DI	Diner's Club card
DL	Driver's license number
DN	Doctor number
DR	Donor Registration Number

Value	Description
DS	Discover Card
EI	Employee number
EN	Employer number
FI	Facility ID
GI	Guarantor internal identifier
GN	Guarantor external identifier
HC	Health Card Number
JHN	Jurisdictional health number (Canada)
LN	License number
LR	Local Registry ID
MA	Medicaid number
MC	Medicare number
MCN	Microchip Number
MR	Medical record number
MS	MasterCard
NE	National employer identifier
NH	National Health Plan Identifier
NI	National unique individual identifier
NNxxx	National Person Identifier where the xxx is the ISO table 3166 3-character (alphabetic) country code
NPI	National provider identifier
PEN	Pension Number
PI	Patient internal identifier
PN	Person number
PRN	Provider number
PT	Patient external identifier
RR	Railroad Retirement number
RRI	Regional registry ID
SL	State license
SR	State registry ID
SS	Social Security number
U	Unspecified
UPIN	Medicare/HCFA's Universal Physician Identification numbers
VN	Visit number
VS	VISA
WC	WIC identifier
WCN	Workers' Comp Number

Value	Description
XX	Organization identifier

2.9.12.6 Assigning facility (HD)

```
Subcomponents: <namespace ID (IS)> & < universal ID (ST)> & <universal ID type (ID)>
```

Definition: The place or location identifier where the identifier was first assigned to the patient. This component is not an inherent part of the identifier but rather part of the history of the identifier: as part of this data type, its existence is a convenience for certain intercommunicating systems.

Note: When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID* (referenced by the first sub-component of the HD component), may be re-defined (given a different user-defined table number and name) by the technical committee responsible for that segment.

2.9.12.7 Effective date (DT)

Definition: The first date, if known, on which the identifier is valid and active.

2.9.12.8 Expiration date (DT)

Definition: The last date, if known, on which the identifier is valid and active.

2.9.13 DLN - driver's license number

```
Components:  <license number (ST)> ^ <issuing state, province, country (IS)> ^ <expiration date
             (DT)
```

Definition: This field contains the driver's license information. For state or province refer to official postal codes for that country; for country refer to ISO 3166 for codes.

2.9.13.1 Driver's license number (as ST data type)

This field contains the driver's license number.

2.9.13.2 Issuing state, province, country (IS)

Issuing authority for driver's license. For state or province refer to official postal codes for that country; for country refer to ISO 3166 for codes. (The ISO 3166 table has three separate forms of the country code: HL7 specifies that the 3-character (alphabetic) form be used for the country code.*)* *User-defined Table 0333 - Driver's license issuing authority* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0333 – Driver's license issuing authority

Value	Description
	No suggested values defined

2.9.13.3 Expiration date (DT)

Expiration date (DT) for driver's license.

2.9.14 DR - date/time range

```
Components:  <range start date/time (TS)> ^ <range end date/time (TS)>
```

```
Subcomponents of range start date/time and range stop date/time:
            YYYY[MM[DD[HHMM[SS[.S[S[S[S]]]]]]]][+/-ZZZZ] & <degree of precision>
```

2.9.14.1 Range start date/time (TS)

Definition: The first component contains the earliest date/time (time stamp) in the specified range.

2.9.14.2 Range end date/time (TS)

The second component contains the latest date/time in the specified range. Note that the TS (time stamp) data type allows the specification of precision.

2.9.15 DT - date

```
Format: YYYY[MM[DD]]
```

In prior versions of HL7, this data type was always specified to be in the format YYYYMMDD. In the current and future versions, the precision of a date may be expressed by limiting the number of digits used with the format specification YYYY[MM[DD]]. Thus, YYYY is used to specify a precision of "year," YYYYMM specifies a precision of "month," and YYYYMMDD specifies a precision of "day."

By site-specific agreement, YYYYMMDD may be used where backward compatibility must be maintained.

Examples:

```
|19880704|
|199503|
```

2.9.16 ED - encapsulated data

```
Components: <source application (HD) > ^ <type of data (ID)> ^ <data subtype (ID)> ^ <encoding
            (ID)> ^ <data (ST)>

Subcomponents: <namespace ID (IS)> & < universal ID (ST)> & <universal ID type (ID)>
```

This data type transmits encapsulated data from a source system to a destination system. It contains the identity of the source system, the type of data, the encoding method of the data, and the data itself. This data type is similar to the RP (reference pointer) data type of Section 2.9.37, "RP - reference pointer," except that instead of pointing to the data on another system, it contains the data which is to be sent to that system.

2.9.16.1 Source application (HD)

A unique name that identifies the system which was the source of the data. Identical format and restrictions as in reference pointer (see Section 2.9.37.2, "Application ID (HD)").

2.9.16.2 Type of data (ID)

Identical to "type of data" component in the reference pointer (RP) data type. (See Section 2.9.37.3, "Type of data (ID)").

Refer to *HL7 Table 0191 – Type of referenced data* for valid values.

2.9.16.3 Data subtype (ID)

Identical to "subtype" component in the reference pointer (RP) data type. (See Section 2.9.37.4, "Subtype (ID)").

Refer to *HL7 Table 0291 - Subtype of referenced data* for valid values.

2.9.16.4 Encoding (ID)

The type of encoding, if present, used to represent successive octets of binary data as displayable ASCII characters. Refer to *HL7 Table 0299 - Encoding* for valid values.

HL7 Table 0299 - Encoding

Value	Description
A	No encoding - data are displayable ASCII characters.
Hex	Hexadecimal encoding - consecutive pairs of hexadecimal digits represent consecutive single octets.
Base64	Encoding as defined by MIME (Multipurpose Internet Mail Extensions) standard RFC 1521. Four consecutive ASCII characters represent three consecutive octets of binary data. Base64 utilizes a 65-character subset of US-ASCII, consisting of both the upper and lower case alphabetic characters, digits "0" through "9," "+," "/," and "=.".

Base64 is defined as follows (adapted from MIME Internet standard RFC 1521, which has precedence over this description). Proceeding from left to right across a 24-bit input group (three octets), each 6-bit group is used as an index into an array of 64 printable characters. The character referenced by the index is placed in the encoded string. These characters are shown in *HL7 Table 0290 - MIME base64 encoding characters*, and are selected so as to be universally representable.

Special processing is performed if fewer than 24 bits are available in an input group at the end of data. A full encoding quantum is always completed at the end of data. When fewer than 24 input bits are available in an input group, zero bits are added (on the right) to form an integral number of 6-bit groups.

Output character positions which are not required to represent actual input data are set to the character "=". Since all canonically encoded output is an integral number of octets, only the following cases can arise: (1) the final quantum of input is an integral multiple of 24 bits; here, the final unit of encoded output will be an integral multiple of 4 characters with no "=" padding, (2) the final quantum of input is exactly 8 bits; here, the final unit of encoded output will be two characters followed by two "="padding characters, or (3) the final quantum of input is exactly 16 bits; here, the final unit of encoded output will be three characters followed by one "=" padding character.

HL7 Table 0290 - MIME base64 encoding characters

Value	Code	Value	Code	Value	Code	Value	Code
0	A	17	R	34	l	51	51 z
1	B	18	S	35	j	52	52 0
2	C	19	T	36	k	53	53 1
3	D	20	U	37	l	54	54 2
4	E	21	V	38	m	55	55 3
5	F	22	W	39	n	56	56 4
6	G	23	X	40	o	57	57 5
7	H	24	Y	41	p	58	58 6
8	I	25	Z	42	q	59	59 7
9	J	26	a	43	r	60	60 8
10	K	27	b	44	s	61	61 9
11	L	28	c	45	t	62	62 +

Value	Code	Value	Code	Value	Code	Value	Code
12	M	29	d	46	u	63	63 /
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

The interpretation of the encoded octets by any of the encoding methods, beyond what is either implicit or specified in the represented data type (such as their ordering within 16-bit or 32-bit binary words on the destination application), is determined by the destination application and is beyond the scope of this Standard.

2.9.16.5 Data (ST)

Displayable ASCII characters which constitute the data to be sent from source application to destination application. The characters are limited to the legal characters of the ST data type, as defined in Section 2.9.43, "ST - string data," and, if encoded binary, are encoded according to the method of Section 2.9.16.2, "Type of data (ID)."

If the encoding component (see Section 2.9.16.4, "Encoding (ID)") = 'A' (none), then the data component must be scanned before transmission for HL7 delimiter characters, and any found must be escaped by using the HL7 escape sequences defined in Section 2.10, "Use of escape sequences in text fields." On the receiving application, the data field must be de-escaped after being parsed.

If the encoding component (see Section 2.9.16.4, "Encoding (ID)") does not equal 'A,' then, after encoding, the (encoded) data must be scanned for HL7 delimiter characters, and any found must be escaped by using the HL7 escape sequences. Only then can the component be added to the HL7 segment/message. On the receiving application, the data field must be de-escaped after being parsed out of the message before being decoded. This can be expressed as 'encode', 'escape', parse, 'de-escape', 'decode'.

2.9.17 EI - entity identifier

```
Components: <entity identifier (ST)> ^ <namespace ID (IS)> ^ <universal ID (ST)> ^ < universal ID
            type (ID)>
```

The entity identifier defines a given entity within a specified series of identifiers.

The EI is appropriate for, but not limited to, machine or software generated identifiers. The generated identifier goes in the first component. The remaining components, 2 through 4, are known as the *assigning authority*; they identify the machine/system responsible for generating the identifier in component 1.

The specified series, the *assigning authority*, is defined by components 2 through 4. The assigning authority is of the hierarchic designator (HD) data type, but it is defined as three separate components in the EI data type, rather than as a single component as would normally be the case. This is in order to maintain backward compatibility with the EI's use as a component in several existing data fields. Otherwise, the components 2 through 4 are as defined in Section 2.9.21, "HD - hierarchic designator." Hierarchic designators (HD) are unique across a given HL7 implementation.

2.9.17.1 Entity identifier (ST)

The first component, <entity identifier>, is usually defined to be unique within the series of identifiers created by the <assigning authority>, defined by a hierarchic designator, represented by components 2 through 4. (See Section 2.9.21, "HD - hierarchic designator".)

2.9.17.2 Namespace ID (IS)

See Section 2.9.21.1, “Namespace ID (IS)” for definition.

The assigning authority is a unique identifier of the system (or organization or agency or department) that creates the data. *User-defined Table 0363 – Assigning authority* is used as the HL7 identifier for the user-defined table of values for this component.

Note:	When the HD is used as a part of another data type, in this case as part of the EI data type, this table may be re-defined (given a different user-defined table number and name) by the technical committee responsible for that segment. By site agreement, implementers may continue to use *User-defined Table 0300 – Namespace ID* for the first component

2.9.17.3 Universal ID (ST)

See Section 2.9.21.2, “Universal ID (ST)” for definition.

2.9.17.4 Universal ID type (ID)

Refer to *HL7 Table 0301 - Universal ID type* for valid values. See Section 2.9.21.3, “Universal ID type (ID),” for definition.

2.9.18 FC - financial class

```
Components:  <financial class (IS)> ^ <effective date (TS)>
```

2.9.18.1 Financial class (IS)

This component contains the financial class assigned to a person. *User-defined Table 0064 - Financial class* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0064 – Financial class

Value	Description
	No suggested values defined

2.9.18.2 Effective date (TS)

This component contains the effective date/time of the person’s assignment to the financial class specified in the first component.

2.9.19 FN - family name

```
Components: <surname (ST)> ^ <own surname prefix (ST)> ^ <own surname (ST)> ^ <surname prefix from
            partner/spouse (ST)> ^ <surname from partner/spouse (ST)>
```

This data type allows full specification of the surname of a person. Where appropriate, it differentiates the person's own surname from that of the person's partner or spouse, in cases where the person's name may contain elements from either name. It also permits messages to distinguish the surname prefix (such as "van" or "de") from the surname root.

Note:	Appears ONLY in the PN and other PN-containing data types (PPN, XCN, XPN).

2.9.19.1 Surname (ST)

The atomic element of the person's family name. In most Western usage, this is the person's last name.

2.9.19.2 Own surname prefix (ST)

Internationalization usage for Germanic languages. This component is optional. An example of a <surname prefix> is the “van” in “Ludwig van Beethoven.” Since the <surname prefix> doesn't sort completely alphabetically, it is reasonable to specify it as a separate sub-component of the PN and extended PN data types (XPN and XCN).

Note:	Subcomponents <own surname prefix>, <own surname>, <surname prefix from partner/spouse> and <surname from partner/spouse> decompose complex Germanic names such as “Irma de Jong-van Beethoven”. If these subcomponents are valued, the <surname> subcomponent should still be fully valued for backward compatibility, i.e. ...^de Jong-van Beethoven&de&Jong&van&Beethoven^... Also, for clarity, the <last name prefix> has been renamed to <own surname prefix>.

2.9.19.3 Own surname (ST)

The portion of the surname (in most Western usage, the last name) that is derived from the person's own surname, as distinguished from any portion that is derived from the surname of the person's partner or spouse. This component is optional.

If the person's surname has legally changed to become (or incorporate) the surname of the person's partner or spouse, this is the person's surname immediately prior to such change. Often this is the person's "maiden name".

2.9.19.4 Surname prefix from partner/spouse (ST)

Internationalization usage for Germanic languages. This component is optional. An example of a <surname prefix> is the “van” in “Ludwig van Beethoven.” Since the <surname prefix> doesn't sort completely alphabetically, it is reasonable to specify it as a separate sub-component of the PN and extended PN data types (XPN and XCN).

Note:	Subcomponents <own surname prefix>, <own surname>, <surname prefix from partner/spouse> and <surname from partner/spouse> decompose complex Germanic names such as “Irma de Jong-van Beethoven”. If these subcomponents are valued, the <surname> subcomponent should still be fully valued for backward compatibility, ie. ...^de Jong-van Beethoven&de&Jong&van&Beethoven^... Also, for clarity, the <last name prefix> has been renamed to <own surname prefix>.

2.9.19.5 Surname from partner/spouse (ST)

The portion of the person's surname (in most Western usage, the last name) that is derived from the surname of the person's partner or spouse, as distinguished from the part derived from the person's own surname. This component is optional.

If no portion of the person's surname is derived from the surname of the person's partner or spouse, this component is not valued. Otherwise, if the surname of the partner or spouse has legally changed to become (or incorporate) the person's surname, this is the surname of the partner or spouse immediately prior to such change.

2.9.20 FT - formatted text data

This data type is derived from the string data type by allowing the addition of embedded formatting instructions. These instructions are limited to those that are intrinsic and independent of the circumstances under which the field is being used. The actual instructions and their representation are described later in this chapter. ***The FT field is of arbitrary length (up to 64k)*** and may contain formatting commands enclosed in escape characters. Example:

```
|\.sp\(skip one vertical line)|
```

For additional examples of formatting commands see Section 2.10, “Use of escape sequences in text fields.”

To include alternative character sets, use the appropriate escape sequence. See Section 2.16.9.18, "Character set", and Section 2.16.9.20, "Alternate character set handling."

2.9.21 HD - hierarchic designator

```
Components:  <namespace ID (IS)> ^ <universal ID  (ST)> ^ <universal ID type (ID)>
```

The HD is designed to be more powerful and more general replacement for the application identifier of HL7 versions 2.1 and 2.2. It adds two additional components, the <universal ID> and the <universal ID type> to the former *application ID* (which is renamed more generically to be the *namespace ID*)

The basic definition of the HD is that it identifies an (administrative or system or application or other) entity that has responsibility for managing or assigning a defined set of instance identifiers (such as placer or filler number, patient identifiers, provider identifiers, etc.). This entity could be a particular health care application such as a registration system that assigns patient identifiers, a governmental entity such as a licensing authority that assigns professional identifiers or drivers' license numbers, or a facility where such identifiers are assigned.

In the case where a HD identifies an entity that assigns/creates instance identifiers such as a particular patient registration system, it defines an "assigning authority." In the case where a HD identifies a location where instance identifiers are given out (although they may be created by another entity at another location) such as a particular "department of motor vehicles office location," it defines an "assigning facility." These two different uses of the HD appear in many of the extended data types.

The "assigning authority" defined by the HD is similar in its role to the coding system (and version) part of the coded element data types: both identify a set of more discrete instance identifiers. The difference is that the set of HD-defined discrete instances contain identifiers of "real-world" things such as patient or clinical orders, while the coded element-defined set of discrete instances contains concept identifiers (codes).

The HD is designed to be used either as a local identifier (with only the <namespace ID> valued) or a publicly-assigned identifier, a UID (<universal ID> and <universal ID type> both valued). Syntactically, 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. HDs that have defined third components (defined UID types) must have a second component that is unique within the series of IDs defined by that component.

Note: The HD is used in fields that in earlier versions of HL7 used the IS data type. Thus, a single component HD (only the first component valued) will look like a simple IS data type for older systems expecting a single component in the place of the HD data type.

If the first component for the HD data type is present, the second and third components are optional. If the third component is present, then the second must also be present (although in this case the first is optional). The second and third components must either both be valued (both non-null), or both be not valued (both null).

This means that 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. However, implementers may choose, by site agreement, to specify that if all three components of the HD are valued, the first component defines a member in the set defined by the second and third components.

2.9.21.1 Namespace ID (IS)

User-defined Table 0300 - Namespace ID is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0300 – Namespace ID

Value	Description
	No suggested values defined

Note:	When the HD is used in a given segment (either as a field or as a component of another data type) this table may be re-defined (given a different user-defined table number and name) by the technical committee responsible for that segment.

2.9.21.2 Universal ID (ST)

The HD's second component, <universal ID> (UID), is a string formatted according to the scheme defined by the third component, <universal ID type> (UID type). The UID is intended to be unique over time within the UID type. It is rigorously defined. Each UID must belong to one of the specifically enumerated schemes for constructing UIDs (defined by the UID type). The UID (second component) must follow the syntactic rules of the particular universal identifier scheme (defined by the third component). Note that these syntactic rules are not defined within HL7 but are defined by the rules of the particular universal identifier scheme (defined by the third component).

2.9.21.3 Universal ID type (ID)

The third component governs the interpretation of the second component of the HD. If the third component is a known UID refer to *HL7 Table 0301 - Universal ID type* for valid values, then the second component is a universal ID of that type.

HL7 Table 0301 - Universal ID type

Value	Description
DNS	An Internet dotted name. Either in ASCII or as integers
GUID	Same as UUID.
HCD	The CEN Healthcare Coding Scheme Designator. (Identifiers used in DICOM follow this assignment scheme.)
HL7	Reserved for future HL7 registration schemes
ISO	An International Standards Organization Object Identifier
L,M,N	These are reserved for locally defined coding schemes.
Random	Usually a base64 encoded string of random bits. The uniqueness depends on the length of the bits. Mail systems often generate ASCII string "unique names," from a combination of random bits and system names. Obviously, such identifiers will not be constrained to the base64 character set.
UUID	The DCE Universal Unique Identifier
x400	An X.400 MHS format identifier
x500	An X.500 directory name

Note:	X400, X500, and DNS are not technically universally valid for all time. Names can be de-registered from an existing user and registered to a new user.

Examples:

Universal ID examples with only the 2nd and 3rd components valued:

```
^1.2.344.24.1.1.3^ISO
```

A HD consisting only of an ISO UID.

```
^1.2.34.4.1.5.1.5.1,1.13143143.131.3131.1^ISO
```

The syntax of the second component is defined by the ISO standard for object identifiers, not by HL7 (for which the second component is of the ST data type). Thus the periods (".") and comma (",") in the second component are part of the ISO syntax, but are legal by the definition of the HL7 ST data type.

```
^14344.14144321.4122344.14434.654^GUID
^falcon.iupui.edu^DNS
```

An internet example

```
^40C983F09183B0295822009258A3290582^RANDOM
```

An example of a RANDOM UID

Local examples:

```
LAB1
```

Local use only: a HD that looks like an IS data type

```
PathLab^PL.UCF.UC^L
```

The 'PathLab' application is identified by the namespace component but it is also identified by the 2nd and 3rd components, (i.e., by the locally-defined UID system "L"). The two identifiers are equivalent. This is a more complex HD in which the middle component, which is locally defined, is itself structured. As with the ISO example above, the middle component's structure is not defined by HL7 but by the site according to its own needs: the only requirement is that the middle component's structure is allowed by the HL7 string (ST) data type.

```
RX.PIMS.SystemB.KP.CA.SCA
```

Local use only: a HD that looks like an IS data type. Again, note that the syntax of the first component is not defined by HL7 but by the site according to its own needs: the only requirement is that the first component's structure is allowed by the HL7 string (ST) data type, which is used for values by the IS data type.

```
^RX.PIMS.SystemB.CA.SCA^M
```

An alternate way to encode the previous example, illustrating the use of the third component value of "M" (see above *HL7 Table 0301*) to identify a locally-defined identifier set. The second component has the same value as the previous example but is now defined to be a member of a set of allowable values defined by a site for the identifier set "M".

Examples containing both local and universal ID types:

```
LAB1^1.2.3.3.4.6.7^ISO
```

A HD with an ISO "object Identifier" as a UID and a locally defined system name. Both the first component and the second and third (taken together) refer to the same entity. This example shows that the local value and the universal ID value may be transmitted with a single HD field.

2.9.22 ID - coded value for HL7 defined tables

The value of such a field follows the formatting rules for an ST field except that it is drawn from a table of legal values. There shall be an HL7 table number associated with ID data types. An examples of an ID field is *OBR-25-result status*. This data type should be used only for HL7 tables (see Section 2.7.6, "Table"). The reverse is not true, since in some circumstances it is more appropriate to use the CE data type for HL7 tables.

2.9.23 IS - coded value for user-defined tables

The value of such a field follows the formatting rules for a ST field except that it is drawn from a site-defined (or user-defined) table of legal values. There shall be an HL7 table number associated with IS data types. An example of an IS field is the *Event reason code* defined in Section 3.3.1.4, "Event reason code."

This data type should be used only for user-defined tables (see Section 2.7.6, “Table”). The reverse is not true, since in some circumstances, it is more appropriate to use the CE data type for user-defined tables.

2.9.24 JCC - job code/class

```
Components: <job code (IS)> ^ <job class (IS)>
```

2.9.24.1 Job code (IS)

This component contains the person’s job code. *User-defined Table 0327 - Job code* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0327 – Job code

Value	Description
	No suggested values defined

2.9.24.2 Job class (IS)

This component contains the person’s employee classification. *User-defined Table 0328 - Employee classification* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0328 – Employee classification

Value	Description
	No suggested values defined

2.9.25 MA - multiplexed array

```
Components:  <sample 1 from channel 1 (NM)> ^ <sample 1 from channel 2 (NM)> ^ <sample 1 from
             channel 3 (NM)> ...~<sample 2 from channel 1 (NM)> ^ <sample 2 from channel 2 (NM)> ^
             <sample 2 from channel 3 (NM)> ...~
              ...
```

This data type is used to represent channel-multiplexed waveform data, (e.g., the digitized values from an analog-to-digital converter or other digital data source). Refer to Chapter 7, Section 7.14.1.2, “MA - multiplexed array,” for a complete description of this data type.

2.9.26 MO - money

```
Components:  <quantity (NM)> ^ <denomination (ID)>
```

Note: Intent is that it appear only as a component of data type CP. Used independently in chapter 8, section 8.10.3.

2.9.26.1 Quantity (NM)

The first component is a quantity.

2.9.26.2 Denomination (ID)

The second component is the denomination in which the quantity is expressed. The values for the denomination component are those specified in ISO-4217. If the denomination is not specified, *MSH-17-country code* is used to determine the default. Example:

```
|99.50^USD|
```

where USD is the ISO 4217 code for the U.S. American dollar.

2.9.27 NA - numeric array

This data type is used to represent a series (array) of numeric values, each one having a data type of NM. Refer to Chapter 7, Section 7.14.1.1, "NA - numeric array," for a complete description of this data type.

2.9.28 NM - numeric

A number represented as a series of ASCII numeric characters consisting of an optional leading sign (+ or -), the digits and an optional decimal point. In the absence of a sign, the number is assumed to be positive. If there is no decimal point the number is assumed to be an integer. Examples:

```
|999|
|-123.792|
```

Leading zeros, or trailing zeros after a decimal point, are not significant. For example, the following two values with different representations, "01.20" and "1.2", are identical. Except for the optional leading sign (+ or -) and the optional decimal point (.), no non-numeric ASCII characters are allowed. Thus, the value <12 should be encoded as a structured numeric (SN) (preferred) or as a string (ST) (allowed, but not preferred) data type.

2.9.29 PL - person location

```
Components:  <point of care  (IS)> ^ <room (IS)> ^ <bed (IS)> ^ <facility (HD)> ^ < location
             status  (IS )> ^ <person location type (IS)> ^ <building (IS )> ^ <floor (IS)> ^ <lo-
             cation description (ST)>
```

Note:	This data type contains several location identifiers that should be thought of in the following order from the most general to the most specific: facility, building, floor, point of care, room, bed. Additional data about any location defined by these components can be added in the following components: person location type, location description and location status.

This data type is used to specify a patient location within a healthcare institution. Which components are valued depends on the needs of the site. For example for a patient treated at home, only the person location type is valued. It is most commonly used for specifying patient locations, but may refer to other types of persons within a healthcare setting.

Example: Nursing Unit

A nursing unit at Community Hospital: 4 East, room 136, bed B

```
4E^136^B^CommunityHospital^^N^^^
```

Example: Clinic

A clinic at University Hospitals: Internal Medicine Clinic located in the Briones building, 3rd floor.

```
InternalMedicine^^^UniversityHospitals^^C^Briones^3^
```

Example: Home

The patient was treated at his home.

```
^^^^^H^^^
```

2.9.29.1 Point of care (IS)

Conditional on person location type (e.g., nursing unit or department or clinic). After floor, most general patient location designation. *User-defined Table 0302 - Point of care* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0302 – Point of care

Value	Description
	No suggested values defined

2.9.29.2 Room (IS)

Patient room. After point of care, most general person location designation. *User-defined Table 0303 - Room* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0303 – Room

Value	Description
	No suggested values defined

2.9.29.3 Bed (IS)

Patient bed. After room, most general person location designation. *User-defined Table 0304 - Bed* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0304 – Bed

Value	Description
	No suggested values defined

2.9.29.4 Facility (HD)

Subject to site interpretation but generally describes the highest level physical designation of an institution, medical center or enterprise. Most general person location designation.

(See Section 2.9.21, “HD - hierarchic designator”) for discussion of data type.

Note: When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID* (referenced by the first sub-component of the HD component) may be redefined (given a different user-defined table number and name) by the technical committee responsible for that segment.

2.9.29.5 Location status (IS)

Location (e.g., Bed) status. *User-defined Table 0306 - Location status* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0306 – Location status

Value	Description
	No suggested values defined

2.9.29.6 Person location type (IS)

Person location type is the categorization of the person’s location defined by facility, building, floor, point of care, room or bed. Although not a required field, when used, it may be the only populated field. Usually

includes values such as nursing unit, department, clinic, SNF, physician's office. *User-defined Table 0305 - Person location type* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0305 – Person location type

Value	Description
C	Clinic
D	Department
H	Home
N	Nursing Unit
O	Provider's Office
P	Phone
S	SNF

2.9.29.7 Building (IS)

After facility, most general person location designation. *User-defined Table 0307 - Building* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0307 – Building

Value	Description
	No suggested values defined

2.9.29.8 Floor (IS)

After building, most general person location designation. *User-defined Table 0308 - Floor* is used as the HL7 identifier for the user-defined table of values for this component.

User-defined Table 0308 – Floor

Value	Description
	No suggested values defined.

2.9.29.9 Location description (ST)

A free text description of the location.

2.9.30 PN - person name

```
Components:  <family name (FN) > ^ <given name (ST)> ^ < second and further given names or ini-
             tials thereof (ST)>  ^ <suffix (e.g., JR or III) (ST)> ^ <prefix (e.g., DR) (ST)> ^
             <degree (e.g., MD) (IS)>

Subcomponents of family name:  <surname (ST)> &<own surname prefix (ST)> &<own surname (ST)>
             &<surname prefix from partner/spouse (ST)> &<surname from partner/spouse (ST)>
```

Note: Replaced by XPN data type as of v 2.3

Length: 48

This data type includes multiple free text components. The sending system may send upper- and lowercase or all uppercase. The receiving system may convert to all uppercase if required. Example:

```
|SMITH^JOHN^J^III^DR^PHD|
```

2.9.30.1 Family name (FN)

This component allows full specification of the surname of a person. Where appropriate, it differentiates the person's own surname from that of the person's partner or spouse, in cases where the person's name may contain elements from either name. It also permits messages to distinguish the surname prefix (such as "van" or "de") from the surname root. See section 2.9.19

2.9.30.2 Given name (ST)

First name.

2.9.30.3 Second and further given names or initials thereof (ST)

Multiple middle names may be included by separating them with spaces.

2.9.30.4 Suffix (ST)

Used to specify a name suffix (e.g., Jr. or III).

2.9.30.5 Prefix (ST)

Used to specify a name prefix (e.g., Dr.).

2.9.30.6 Degree (IS)

Used to specify an educational degree (e.g., MD). Refer to *User-defined Table 0360 – Degree* for suggested values.

2.9.30.6.1 Internationalization note

In countries using ideographic or syllabic (phonetic) character sets, it is sometimes necessary to send the name in one or both of these formats, as well as an alphabetic format. The switching between the different character sets can be accomplished using a character set such as JIS X 0202 - ISO 2022 which provides an escape sequence for switching among different character sets and among single-byte and multi-byte character representations. When the name field is repeated, the different repetitions of the name may be represented by these different character sets. The details are as follows. (See also Section 2.9.2, “Escape sequences supporting multiple character sets for PN, XPN, XCN, XON, XAD, FT, ST and TX data types.”)

HL7 supports the following standards for Japanese characters:

JIS X 0201 for ISO-IR 13 (Japanese Katakana)

JIS X 0201 for ISO-IR 14 (Japanese Romaji)

JIS X 0208 for ISO-IR 87 (Japanese Kanji, Hiragana and Katakana)

JIS X 0212 for ISO-IR 159 (supplementary Japanese Kanji)

HL7 supports the following standards for European characters:

ISO 8859 (1-9) for ISO-IR 100, 101, 109, 110, 144,127, 126, 138 and 148.

Character sets are referenced in HL7 as ASCII, 8859/1,.8859/2, ISO IR14, ISO IR87, and ISO IR159. DICOM uses codes laid out in ISO 2375, of the form 'ISO-IR xxx'. HL7 supports this naming as well, to facilitate interoperability.

HL7 uses the Basic G0 Set of the International Reference Version of ISO 646:1990 (ISO IR-6) as the default character repertoire for character strings. This is a single-byte character set, identical to ASCII.

Each repetition of a PN, XPN, XON, XCN, or XAD field is assumed to begin with the default character set. If another character set is to be used, the HL7 defined escape sequence used to announce that character set must be at the beginning of the repetition, and the HL7 defined escape sequence used to start the default character set must be at the end of the repetition. Note also that several character sets may be intermixed within a single repetition as long as the repetition ends with a return to the default character set.

An application must specify which character sets it supports in the field "MSH-18 Character Sets" and which character set handling scheme it supports in the field *MSH-20-Alternate character set handling scheme*. It is assumed that the sending and receiving applications are aware of how to map character set names (i.e., ISO-IR xxx) to escape sequences.

For example, in many Japanese messages there is a mix of Romaji (i.e., Roman characters), Katakana (phonetic representation of foreign words), Hiragana (phonetic representation of Japanese words) and Kanji (pictographs). Such a message would require 4 character sets be specified in the MSH.

2.9.30.7 References for internationalization of name

1. "Understanding Japanese Information Processing" by Ken Lunde, O'Reilly Press
2. "DICOM Supplement 9 : Multi-Byte Character Set Support", ACR-NEMA
3. ANSI X3.4:1986 — ASCII character set
4. ISO 646:1990 — Information Processing - ISO 7-bit coded character set for information interchange
5. ISO/IEC 2022:1994 — Information Technology - Character code structure and extension techniques
6. ISO 2375:1986 — Data Processing - Procedure for the registration of escape sequences
7. ISO 6429:1990 — Information Processing - Control functions for 7-bit and 8-bit coded character sets
8. ISO 8859 (1-9) — Information Processing - 8-bit single-byte coded graphic character sets - parts 1-9
9. ENV 41 503:1990 — Information systems interconnection - European graphic character repertoires and their coding
10. ENV 41 508:1990 — Information systems interconnection - East European graphic character repertoires and their coding
11. JIS X 0201-1976 — Code for Information Exchange
12. JIS X 0212-1990 — Code of the supplementary Japanese Graphic Character set for information interchange

13. JIS X 0208-1990 Code for the Japanese Graphic Character set for information interchange

14. RFC 1468 Japanese Character Encoding for Internet Messages

This approach is in harmony with DICOM.

Character Repertoires supported by DICOM are defined in Part 5, section 62E1, of Supplement 9. It says, "Values that are text or character strings can be composed of Graphic and Control Characters. The Graphic Character set, independent of its encoding, is referred to as a Character Repertoire. Depending on the native context in which Application Entities wish to exchange data using the DICOM standard, different character repertoires will be used. The Character Repertoires supported by DICOM are defined in ISO 8859."

In addition, DICOM supports the following Character Repertoires for the Japanese Language:

JIS X 0201-1976 - Code for Information Exchange

JIS X 0208-1990 - Code for the Japanese Graphic Character set for information interchange

JIS X 0212-1990 - Code of the supplementary Japanese Graphic Character set for information interchange

2.9.31 PPN - performing person time stamp

```
Components: <ID number (ST)> ^ <family name (FN)> ^ <given name (ST) ^ <second and further given
            names or initials thereof (ST)> ^ <suffix (e.g., JR or III) (ST)> ^ <prefix (e.g.,
            DR) (ST)> ^ <degree (e.g., MD) (IS)> ^ <source table (IS)> ^ <assigning authority
            (HD)> ^ <name type code(ID)>  ^ <identifier check digit (ST)> ^ <code identifying the
            check digit scheme employed (ID )> ^ <identifier type code (IS)> ^ <assigning facil-
            ity (HD)> ^ < date/time action performed (TS)> ^ <name representation code (ID)> ^
            <name context (CE)> ^ <name validity range (DR)> ^ <name assembly order (ID)>

Subcomponents of family name:  <surname (ST)> &<own surname prefix (ST)> &<own surname (ST)>
            &<surname prefix from partner/spouse (ST)> &<surname from partner/spouse (ST)>

Subcomponents of name context: <identifier (ST)> & <text (ST)> & <name of coding system (IS)> &
            <alternate identifier (ST)> & <alternate text (ST)> & <name of alternate coding sys-
            tem (IS)>

Subcomponents of name validity range:  <date range start date/time (TS)> & <date range end
            date/time (TS)>
```

Length: 250

This data type is the equivalent of an XCN data type joined with a TS data type. However, since HL7 does not support subcomponents in Version 2.3, the XCN data type has been flattened.

2.9.31.1 ID number (ST)

Coded ID according to a user-defined table, defined by the 8th component. If the first component is present, either the source table or the assigning authority must be valued.

2.9.31.2 Family name (FN)

This component allows full specification of the surname of a person. Where appropriate, it differentiates the person's own surname from that of the person's partner or spouse, in cases where the person's name may contain elements from either name. It also permits messages to distinguish the surname prefix (such as "van" or "de") from the surname root. See section 2.9.19.

2.9.31.3 Given name (ST)

First name.

2.9.31.4 Second and further given names or initials thereof (ST)

Multiple middle names may be included by separating them with spaces.

2.9.31.5 Suffix (ST)

Used to specify a name suffix (e.g., Jr. or III).

2.9.31.6 Prefix (ST)

Used to specify a name prefix (e.g., Dr.).

2.9.31.7 Degree (IS)

Used to specify an educational degree (e.g., MD). Refer to *User-defined Table 0360 – Degree* for suggested values.

2.9.31.8 Source table (IS)

User-defined Table 0297 - CN ID *source* is used as the HL7 identifier for the user-defined table of values for this component. Used to delineate the first component.

2.9.31.9 Assigning authority (HD)

The assigning authority is a unique identifier of the system (or organization or agency of department) that creates the data. It is a HD data type. *User-defined Table 0363 – Assigning authority* is used as the HL7 identifier for the user-defined table of values for the first sub-component of the HD component, <namespace ID>.

Note:	When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID* (referenced by the first sub-component of the HD component) may be re-defined (given a different user-defined table number and name) by the technical committee responsible for that segment. By site agreement, implementers may continue to use *User-defined Table 0300 – Namespace ID* for the first sub-component.

2.9.31.10 Name type code (ID)

A code that represents the type of name. Refer to *HL7 Table 0200 - Name type* for valid values (see Section 2.9.55, “XPN - extended person name”).

2.9.31.11 Identifier check digit (ST)

The check digit in this data type is not an add-on produced by the message processor. It is the check digit that is part of the identifying number used in the sending application. If the sending application does not include a self-generated check digit in the identifying number, this component should be valued null.

2.9.31.12 Code identifying the check digit scheme employed (ID)

Refer to *HL7 Table 0061 - Check digit scheme* for valid values.

2.9.31.13 Identifier type code (IS)

A code corresponding to the type of identifier. In some cases, this code may be used as a qualifier to the "Assigning authority" component. Refer *to HL7 Table 0203 - Identifier type* for suggested values.

2.9.31.14 Assigning facility (HD)

The place or location identifier where the identifier was first assigned to the patient. This component is not an inherent part of the identifier but rather part of the history of the identifier: as part of this data type, its existence is a convenience for certain intercommunicating systems.

Note:	When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID* (referenced by the first sub-component of the HD component) may be re-defined (given a different user-defined table number and name) by the technical committee responsible for that segment.

2.9.31.15 Date/time action performed (TS)

This component describes when the activity was performed.

Note:	If this field is not null, both the performing person and the time stamp must be valued.

2.9.31.16 Name representation code (ID)

Different name/address types and representations of the same name/address should be described by repeating of this field, with different values of the Name/Address Type and/or Name/Address Representation component.

Note:	This new component remains in "alphabetic" representation with each repetition of the field using these data types. I.e. even though the name may be represented in an ideographic character set, this component will remain represented in an alphabetic character set.

HL7 Table 0465 - Name/address representation

Value	Description
I	Ideographic (i.e., Kanji)
A	Alphabetic (i.e., Default or some single-byte)
P	Phonetic (i.e., ASCII, Katakana, Hiragana, etc.)

In general this component provides an indication of the representation provided by the data item. It does not necessarily specify the character sets used. Thus, even though the representation might provide an indication of what to expect, the sender is still free to encode the contents using whatever character set is desired. This component provides only hints for the receiver, so it can make choices regarding what it has been sent and what it is capable of displaying.

2.9.31.17 Name context (CE)

```
Subcomponents of name context:  <identifier (ST)> & <text (ST)> & <name of coding system (IS)> &
            <alternate identifier (ID)> & <alternate text (ST)> & <name of alternate coding sys-
            tem (IS)>
```

This component is used to designate the context in which a name is used. The main use case is in Australian healthcare: indigenous patients who prefer to use different names when attending different healthcare institutions. Another use case occurs in the US where health practitioners can be licensed under slightly different names and the reporting of the correct name is vital for administrative purposes. Refer to chapter 3, section 3.4.2.6 for more detailed information on how to use this table. Refer to *User-defined table 0448 – Name context* for suggested values;

User-defined Table 0448 – Name context

Value	Description
	No suggested values defined

2.9.31.18 Name validity range (DR)

This component contains the start and end date/times which define the period during which this name was valid. See section 2.9.14, “DR - date/time range” for description of subcomponents.

2.9.31.19 Name assembly order (ID)

A code that represents the preferred display order of the components of this person name. Refer to *HL7 Table 0444 – Name assembly order* for valid values.

HL7 Table 0444 – Name assembly order

Value	Description
G	Prefix Given Middle Family Suffix
F	Prefix Family Middle Given Suffix

2.9.32 PT - processing type

```
Components: <processing ID (ID)> ^ <processing mode (ID)>
```

This data type indicates whether to process a message as defined in HL7 Application (level 7) Processing rules.

2.9.32.1 Processing ID (ID)

A value that defines whether the message is part of a production, training, or debugging system. Refer to *HL7 Table 0103 - Processing ID* for valid values.

2.9.32.2 Processing mode (ID)

A value that defines whether the message is part of an archival process or an initial load. Refer to *HL7 Table 0207 - Processing mode* for valid values.

2.9.33 QIP - query input parameter list

Components: <segment field name (ST) > ^ <value1 (ST) & value2 (ST) & value3 (ST) ...>

Example:

```
|@PID.5.1^EVANS|
```

Definition: This field contains the list of parameter names and values to be passed to the stored procedure.

2.9.33.1 Segment field name (ST)

This component contains the segment field name.

Naming conventions:

Segment field names are designated by the “@” symbol concatenated with the HL7 segment ID followed by the sequence number for the field separated by a period (see sections 2.6, “SEGMENTS” and 2.7.1,

"Position (sequence within the segment)" for a definition of segment ID and sequence number). If the field is divided into components, the designation may be suffixed with ".nn," to identify a particular component (a suffix of ".3" indicates the third component of the field); otherwise, the whole field is assumed. If the field is further divided into subcomponents, the designation is suffixed with ".nn.mm," which identifies the component and subcomponent requested by relative position.

Site-specific segment field names may be used**.** In this case, the site-specific segment ID (if the field is not being added to an existing HL7 segment) and the sequence number must be defined so that they do not conflict with existing HL7 segment IDs and field sequence numbers.

Values for this field are defined in the function-specific chapters of this specification.

Note:	If the "@" is being used as one of the delimiter characters defined in MSH-2-encoding characters, it must be "escaped." (See Section 2.10.1, "Formatting codes".)

2.9.33.2 Value1 & value2 & value3 (ST)

This component contains the field value or values in the form "value1& value2 & value3…"

A single valued parameter contains only a single subcomponent in the second component: thus no subcomponent delimiters are needed (e.g., `<segment field name> ^ <value>`). A simple list of values (i.e., a one-dimensional array) may be passed instead of a single value by separating each value with the subcomponent delimiter: ,"`<segment field name>` ^ `<value1 & value2 &…>`"

2.9.34 QSC - query selection criteria

```
Components: <segment field name(ST)> ^ <relational operator (ID)> ^ <value (ST)> ^ <relational
            conjunction (ID)>
```

Example:

```
|@PID.5.1^EQ^EVANS|
```

Definition: This field indicates the conditions that qualify the rows to be returned in the query response. (This field conveys the same information as the "WHERE" clause in the corresponding SQL expression of the query, but is formatted differently.)

2.9.34.1 Segment field name (ST)

The name of the field that is participating as a qualifier (usually the "key"). Refer to Section 2.9.33.1, "Segment field name (ST)," for segment field name conventions.

2.9.34.2 Relational operator (ID)

Refer to *HL7 Table 0209 - Relational operator* for valid values.

HL7 Table 0209 - Relational operator

Relational operator	Value
EQ	Equal
NE	Not Equal
LT	Less than
GT	Greater than
LE	Less than or equal
GE	Greater than or equal

Relational operator	Value
CT	Contains
GN	Generic

2.9.34.3 Value (ST)

The value to which the field will be compared.

2.9.34.4 Relational conjunction (ID)

Refer *to HL7 Table 0210 - Relational conjunction* for valid values. The relational conjunction is defined as follows: If more than one comparison is to be made to select qualifying rows, a conjunction relates this repetition of the field to the next.

HL7 Table 0210 - Relational conjunction

Relational conjunction	Note
AND	Default
OR	

- When applied to strings, the relational operators LT, GT, LE, and GE imply an alphabetic comparison.
- A "generic" comparison selects a record for inclusion in the response when the beginning of the designated field matches the select string.
- Where a repeating field is specified as an operand, a match on any instance of that field qualifies the row for inclusion in the response message.
- AND takes precedence over OR. More sophisticated precedence rules require that the query be expressed as an embedded query language message or a stored procedure query message (see chapter 5)

2.9.35 RCD - row column definition

```
Components: &lt;segment field name (ST)&gt; ^ &lt;HL7 data type (ID)&gt; ^ &lt;maximum column width (NM)&gt;
```

Example: This defines a column containing the value of the "last name" component of PID-5, expressed as a ST data type with a maximum width of 20.

```
|@PID.5.1^ST^20|
```

Definition: This specifies the format of a column in terms of a segment field name, a data type, and a maximum length. It consists of three components:

2.9.35.1 Segment field name (ST)

The HL7 segment field name, which identifies the field occupying the column. (Refer to Section 2.9.33.1, "Segment field name (ST)," for segment field name definition conventions.)

2.9.35.2 HL7 data type (ID)

The two or three character HL7 data type. *Refer to HL7 Table 0440 – Data Types* for valid values.

2.9.35.3 Maximum column width (NM)

The maximum width of the column, as dictated by the responding system. (This may vary from the HL7-defined maximum field length.)

2.9.36 RI - repeat interval

```
Components:  <repeat pattern (IS)> ^ <explicit time interval (ST)>
```

Definition: This field contains the interval between repeating appointments. The default setting indicates that the appointment should occur once, when the component is not valued. The definition of this field is equivalent to the definition of the Interval component of the Quantity/Timing field given in Chapter 4, Section 4.4.2 “Interval component (CM).”

2.9.36.1 Repeat pattern (IS)

The first component is defined by *User-defined Table 0335 - Repeat pattern*. See Section 4.3.2.1 “Repeat pattern,” for further details.

2.9.36.2 Explicit time interval (ST)

The second component explicitly lists the actual times referenced by the code in the first subcomponent, in the following format: HHMM,HHMM,HHMM,.… This second subcomponent will be used to clarify the first subcomponent in cases where the actual administration times vary within an institution. See Section 4.4.2.2, “Explicit time interval,” for further details.

2.9.37 RP - reference pointer

```
Components:  <pointer (ST) > ^ < application ID (HD)> ^ <type of data (ID)> ^ <subtype (ID)>
```

This data type transmits information about data stored on another system. It contains a reference pointer that uniquely identifies the data on the other system, the identity of the other system, and the type of data.

2.9.37.1 Pointer (ST)

A unique key assigned by the system that stores the data. The key, which is a ST data type, is used to identify and access the data.

2.9.37.2 Application ID (HD)

```
Subcomponents: <namespace ID (IS)> & < universal ID (ST)> & <universal ID type (ID)>
```

A unique designator of the system that stores the data. It is a HD data type (See Section 2.9.21, “HD - hierarchic designator”). Application ID’s must be unique across a given HL7 implementation.

2.9.37.3 Type of data (ID)

An ID data type that declares the general type of data. Refer to *HL7 Table 0191- Type of referenced data* for valid values.

HL7 Table 0191 - Type of referenced data

Value	Description
AP	Other application data, typically uninterpreted binary data (HL7 V2.3 and later)
AU	Audio data (HL7 V2.3 and later)

Value	Description
FT	Formatted text (HL7 V2.2 only)
IM	Image data (HL7 V2.3 and later)
multipart	MIME multipart package
NS	Non-scanned image (HL7 V2.2 only)
SD	Scanned document (HL7 V2.2 only)
SI	Scanned image (HL7 V2.2 only)
TEXT	Machine readable text document (HL7 V2.3.1 and later)
TX	Machine readable text document (HL7 V2.2 only)

2.9.37.4 Subtype (ID)

An ID data type declaring the format for the data of subcomponent <main type>. Refer to *HL7 Table 0291 - Subtype of referenced data* for valid values.

HL7 Table 0291 - Subtype of referenced data

Value	Description
BASIC	ISDN PCM audio data
DICOM	Digital Imaging and Communications in Medicine
FAX	Facsimile data
GIF	Graphics Interchange Format
HTML	Hypertext Markup Language
JOT	Electronic ink data (Jot 1.0 standard)
JPEG	Joint Photographic Experts Group
Octet-stream	Uninterpreted binary data
PICT	PICT format image data
PostScript	PostScript program
RTF	Rich Text Format
SGML	Standard Generalized Markup Language (HL7 V2.3.1 and later)
TIFF	TIFF image data
x-hl7-cda-level-one	HL7 Clinical Document Architecture Level One document
XML	Extensible Markup Language (HL7 V2.3.1 and later)

2.9.37.5 Type-subtype combinations

Possible subtypes are specific to main types (though in principle the same subtype could be used for more than one main type), and so are defined under their main types.

Additional subtypes may be added to this Standard. In addition, private, non-standard subtypes may be defined by agreement between cooperating parties. All private, non-standard subtypes should begin with the letter **Z** to distinguish them from the standard subtypes.

2.9.37.5.1 Image subtypes

```
TIFF = TIFF image data
```

TIFF (Tagged Image File Format) is one of the common formats for scanned images. Its first version was developed in 1986 by Aldus Corporation as a standard for encoding scanned images. The official version of the TIFF standard is now maintained by Adobe Corporation. TIFF format is specified in the document "TIFF, Revision 6.0." Adobe Systems Incorporated, 1585 Charleston Road, P.O. Box 7900, Mountain View, CA 94039-7900. (415) 961-4400

The subtype "TIFF" implies recognition of that trademark and all the rights it entails.

```
PICT = PICT format image data
```

PICT is one of the common formats for scanned images. PICT is a graphics format developed by Apple Computer, Inc., Cupertino, California. PICT format is officially defined in the book set "Inside Macintosh," published by Addison-Wesley Publishing Company, Reading, Massachusetts.

```
DICOM = the Digital Imaging and Communications in Medicine (DICOM) standard
```

DICOM is the format developed jointly by the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) as the standard for interchange of radiological images and ancillary data. It is standardized as NEMA PS3, and is available from: NEMA, 2101 L Street NW, Washington, DC 20037

DICOM specifies a complete communications standard, including a generic messaging service for two-way exchange of imaging-related information between applications, as well as transfer of the actual images. In HL7, the use of DICOM data is limited to images only.

Images in this subtype shall be encoded according to the Generic DICOM File Format defined in DICOM Part 10, Media Storage and File Format (NEMA PS3.10). This shall be in accordance with the Image Information Object Definitions of DICOM Part 3 (NEMA PS3.3), Data Structure and Semantics of DICOM Part 5 (NEMA PS3.5), and the Data Dictionary of DICOM Part 6 (NEMA PS3.6).

The Generic DICOM File Format consists of two parts: a DICOM File Meta Information Header, immediately followed by a DICOM Data Set. The DICOM Data Set contains the image or images specified according to DICOM Part 10. The DICOM File Meta Information Header contains, among other information, a Transfer Syntax UID (Unique Identifier) which completely specifies the encoding of the Data Set according to DICOM Part 5. This encoding defines big endian vs. little endian byte ordering, as well as image compression via the JPEG (Joint Photographics Experts Group) standard (ISO/IS 10918-1 and 10918-2). The transfer syntax of the File Meta Information Header itself is little endian byte ordered, as required by DICOM Part 10.

```
FAX = facsimile data
```

Facsimile data as specified by CCITT standards F1.60, F1.80, F1.82, and F1.84.

```
Jot = electronic ink data, as specified by the Jot 1.0 standard
```

The JOT standard, proposed jointly by Slate Corporation, Microsoft, Apple, Lotus, GO, and General Magic, allows handwritten notes, sketches, signatures and other free-form written data to be transmitted. It is the standard by which portable pen computers or workstations equipped with stylus-input tablets can represent and exchange information.

It represents electronic ink as a series of stylus strokes, and therefore contains necessary information for potential automatic handwriting recognition, which would be lost if converted to other image representa-

tions. It may, however, be readily converted to another image representation for purposes of printing or display.

The JOT 1.0 standard is available from: Software Publishers Association, 1730 M Street Northwest, Suite 700
Washington, DC 20036-4510, (202) 452-1600

2.9.37.5.2 Audio subtypes

```
basic = ISDN PCM audio data
```

Telephone quality audio data, encoded as 8-bit ISDN mu-law Pulse Code Modulation sampled at 8 kHz, according to CCITT Fascicle III.4, Recommendation G.711. This subtype may be used for voice mail messages as well as voice dictation.

2.9.37.5.3 Application subtypes

```
octet-stream = uninterpreted binary data
```

This subtype is for binary data which has none of the other standard formats as given by Section 2.9.37.3, "Type of data (ID)." Its interpretation by the system utilizing the data must be mutually agreed upon by sending and receiving parties.

```
PostScript = PostScript program
```

A PostScript language program typically representing a formatted document for printing on a PostScript printer, or for display on a computer screen via a PostScript interpreter.

PostScript consists of the original specification, PostScript level 1, described in "PostScript Language Reference Manual," Addison-Wesley, 1985, and a more advanced variant, PostScript level 2, described in "PostScript Language Reference Manual," Addison-Wesley, Second Edition, 1990. PostScript is a registered trademark of Adobe Systems, Inc. Use of the subtype "PostScript" implies recognition of that trademark and all the rights it entails.

Other types may be added as needed.

Example:

|1234A321634BC^EFC^SD|

2.9.38 SAD – street address

```
Components: <street or mailing address (ST)> ^ <street name (ST)> ^ <dwelling number (ST)>
```

Note:	Appears ONLY in the XAD data type

2.9.38.1 Street or mailing address (ST)

The street or mailing address of a person or institution. When referencing an institution, this first component is used to specify the institution name. When used in connection with a person, this component specifies the first line of the address.

2.9.38.2 Street name (ST)

2.9.38.3 Dwelling number (ST)

2.9.39 SCV - scheduling class value pair

```
Components:  <parameter class (IS)> ^ <parameter value (ST)>
```

For use only with the scheduling chapter.

Definition: This data type is used to communicate parameters and preferences to the filler application regarding the selection of an appropriate *time slot, resource, location, or filler override criterion* for an appointment.

2.9.39.1 Parameter class (IS)

The first component of this field is a code identifying the parameter or preference being passed to the filler application. Refer to *User-defined table 0294 Time selection criteria parameter class codes* for suggested values.

User-defined Table 0294 - Time selection criteria parameter class codes

Parameter Class	Description: Valid Values
Prefstart	The preferred start time for the appointment request, service or resource. Any legal time specification in the format HHMM, using 24-hour clock notation
Prefend	The preferred end time for the appointment request, service or resource. Any legal time specification in the format HHMM, using 24-hour clock notation
Mon	An indicator that Monday is or is not preferred for the day on which the appointment will occur. OK = Preferred appointment day NO = Day is not preferred
Tue	An indicator that Tuesday is or is not preferred for the day on which the appointment will occur. OK = Preferred appointment day NO = Day is not preferred
Wed	An indicator that Wednesday is or is not preferred for the day on which the appointment will occur. OK = Preferred appointment day NO = Day is not preferred
Thu	An indicator that Thursday is or is not preferred for the day on which the appointment will occur. OK = Preferred appointment day NO = Day is not preferred
Fri	An indicator that Friday is or is not preferred for the day on which the appointment will occur. OK = Preferred appointment day NO = Day is not preferred
Sat	An indicator that Saturday is or is not preferred for the day on which the appointment will occur. OK = Preferred appointment day NO = Day is not preferred
Sun	An indicator that Sunday is or is not preferred for the day on which the appointment will occur. OK = Preferred appointment day NO = Day is not preferred

2.9.39.2 Parameter value (ST)

The second component is the actual data value for that parameter.

For example, if a filler application allows preference parameters to be passed to specify a preferred start time, a preferred end time, and preferred days of the week for the appointment, it may define the following parameter class codes and valid data sets.

2.9.40 SI - sequence ID

A non-negative integer in the form of a NM field. The uses of this data type are defined in the chapters defining the segments and messages in which it appears.

2.9.41 SN - structured numeric

```
Components:  <comparator (ST)> ^ <num1 (NM)> ^ <separator/suffix (ST)> ^ <num2 (NM)>
```

The structured numeric data type is used to unambiguously express numeric clinical results along with qualifications. This enables receiving systems to store the components separately, and facilitates the use of numeric database queries. The corresponding sets of values indicated with the <comparator> and <separator/suffix> components are intended to be the authoritative and complete set of values. If additional values are needed for the <comparator> and <separator/suffix> components, they should be submitted to HL7 for inclusion in the Standard.

If <num1> and <num2> are both non-null, then the separator/suffix must be non-null. If the separator is "-", the data range is inclusive; e.g., <num1> - <num2> defines a range of numbers *x*, such that: <num1> <=*x*<= <num2>.

2.9.41.1 Comparator (ST)

Defined as greater than, less than, greater than or equal, less than or equal, equal, and not equal, respectively (= ">" or "<" or ">=" or "<=" or "=" or "<>"

If this component is not valued, it defaults to equal ("=").

2.9.41.2 Num1 (NM)

A number.

2.9.41.3 Separator/suffix (ST)

"-" or "+" or "/" or "." or ":"

Examples:

```
|>^100|           (greater than 100)
|^100^-^200|      (equal to range of 100 through 200)
|^1^:^228|    (ratio of 1 to 128, e.g., the results of a serological test)
|^2^+|        (categorical response, e.g., occult blood positivity)
```

2.9.41.4 Num2 (NM)

A number or null depending on the measurement.

2.9.42 SRT – sort order

```
Components: <sort-by field(ST)> ^ <sequencing (ID)>
```

Specifies those parameters by which the response will be sorted and by what method.

Example: In a tabular response query, where the return data is known by column name, the SRT might look like

|LastName^A~FirstName^A|Example: In a segment response query, where the return data is known by segment and offset, the SRT field would use segment field name notation,

```
|PID.3.1^A~PID.3.2|
```

2.9.42.1 Sort-by field (ST)

Identifies the field by which the response will be sorted. In a tabular response , this will be the column name to sort by. In the Segment Pattern and the Display Response, this will be the segment field name to sort by. (see QIP in Section 2.9.33.1, “Segment field name (ST)” for segment field name definition.)

See Chapter 5, "Query", for a complete discussion of queries and their responses.

2.9.42.2 Sequencing (ID)

Identifies how the field or parameter will be sorted; and, if sorted, whether the sort will be case sensitive (the default) or not. Refer *to HL7 Table 0397 – Sequencing* for valid values

HL7 Table 0397 – Sequencing

Value	Description
A	Ascending
AN	Ascending, case insensitive
D	Descending
DN	Descending, case insensitive
N	None

2.9.43 ST - string data

String data is left justified with trailing blanks optional. Any displayable (printable) ACSII characters (hexadecimal values between 20 and 7E, inclusive, or ASCII decimal values between 32 and 126), except the defined escape characters and defined delimiter characters. Example:

```
|almost any data at all|
```

To include any HL7 delimiter character (except the segment terminator) within a string data field, use the appropriate HL7 escape sequence (see Section 2.10.1, “Formatting codes”).

Usage note: The ST data type is intended for short strings (e.g., less than 200 characters). For longer strings the TX or FT data types should be used (see Sections 2.9.48, “TX - text data” or 2.9.20, “FT - formatted text data”).

Alternate character set note: ST - string data may also be used to express other character sets. See Section 2.15.9.18, "Character set", and Section 2.15.9.20, "Alternate character set handling" for details.

2.9.44 TM - time

```
Format: HH[MM[SS[.S[S[S[S]]]]]][+/-ZZZZ]
```

In prior versions of HL7, this data type was always specified to be in the format HHMM[SS[.SSSS]][+/-ZZZZ] using a 24 hour clock notation. In the current and future versions, the precision of a time may be expressed by limiting the number of digits used with the format specification as shown above. By site-

specific agreement, HHMM[SS[.SSSS]][+/-ZZZZ] may be used where backward compatibility must be maintained.

Thus, HH is used to specify a precision of "hour," HHMM is used to specify a precision of "minute," HHMMSS is used to specify a precision of seconds, and HHMMSS.SSSS is used to specify a precision of ten-thousandths of a second.

In each of these cases, the time zone is an optional component. The fractional seconds could be sent by a transmitter who requires greater precision than whole seconds. Fractional representations of minutes, hours or other higher-order units of time are not permitted.

Note:	The time zone [+/-ZZZZ], when used, is restricted to legally-defined time zones and is represented in HHMM format.

The time zone of the sender may be sent optionally as an offset from the coordinated universal time (previously known as Greenwich Mean Time). Where the time zone is not present in a particular TM field but is included as part of the date/time field in the MSH segment, the MSH value will be used as the default time zone. Otherwise, the time is understood to refer to the local time of the sender. Midnight is represented as 0000. Examples:

|235959+1100|	1 second before midnight in a time zone eleven hours ahead of Universal Coordinated Time (i.e., east of Greenwich).
|0800|	Eight AM, local time of the sender.
|093544.2312|	44.2312 seconds after Nine thirty-five AM, local time of sender.
|13|	1pm (with a precision of hours), local time of sender.

2.9.45 TN - telephone number

For use in the United States and conforming countries, the telephone number is always in the form:

```
Format:  [NN] [(999)]999-9999[X99999][B99999][C any text]
```

Note:	Replaced by XTN data type as of v 2.3

The optional first two digits are the country code. The optional **X** portion gives an extension. The optional **B** portion gives a beeper code. The optional **C** portion may be used for comments like, **After 6:00**. While no explicit limit is placed on the text field, receiving systems may be expected to truncate values that are more than 10 characters long. To accommodate the variability of institutional phone systems, the length of the extension and beeper numbers may be extended by local agreement. Examples:

```
|(415)925-0121X305|
|234-4532CWEEKENDS|
```

2.9.46 TQ - timing quantity

Describes when a service should be performed and how frequently. See Chapter 4 (Section 4.3, "QUANTITY/TIMING (TQ) DEFINITION") for a complete description of this data type.

2.9.47 TS - time stamp

```
Format:  YYYY[MM[DD[HHMM[SS[.S[S[S[S]]]]]]]][+/-ZZZZ]^<degree of precision>
```

Contains the exact time of an event, including the date and time. The date portion of a time stamp follows the rules of a date field and the time portion follows the rules of a time field. The time zone (+/-ZZZZ) is represented as +/-HHMM offset from UTC (formerly Greenwich Mean Time (GMT)), where +0000 or -

0000 both represent UTC (without offset). The specific data representations used in the HL7 encoding rules are compatible with ISO 8824-1987(E).

In prior versions of HL7, an optional second component indicates the degree of precision of the time stamp (Y = year, L = month, D = day, H = hour, M = minute, S = second). This optional second component is retained only for purposes of backward compatibility.

By site-specific agreement, YYYYMMDD[HHMM[SS[.S[S[S[S]]]]]][+/-ZZZZ]^<degree of precision> may be used where backward compatibility must be maintained.

In the current and future versions of HL7, the precision is indicated by limiting the number of digits used, unless the optional second component is present. Thus, YYYY is used to specify a precision of "year," YYYYMM specifies a precision of "month," YYYYMMDD specifies a precision of "day," YYYYMMDDHH is used to specify a precision of "hour," YYYYMMDDHHMM is used to specify a precision of "minute," YYYYMMDDHHMMSS is used to specify a precision of seconds, and YYYYMMDDHHMMSS.SSSS is used to specify a precision of ten thousandths of a second. In each of these cases, the time zone is an optional component. Note that if the time zone is not included, the time-zone defaults to that of the local time zone of the sender. Also note that a TS valued field with the HHMM part set to "0000" represents midnight of the night extending from the previous day to the day given by the YYYYMMDD part (see example below). Maximum length of the time stamp is 26. Examples:

```
|19760704010159-0500|
```

1:01:59 on July 4, 1976 in the Eastern Standard Time zone (USA).

```
|19760704010159-0400|
```

1:01:59 on July 4, 1976 in the Eastern Daylight Saving Time zone (USA).

```
|198807050000|
```

Midnight of the night extending from July 4 to July 5, 1988 in the local time zone of the sender.

```
|19880705|
```

Same as prior example, but precision extends only to the day. Could be used for a birthdate, if the time of birth is unknown.

```
|19981004010159+0100|
```

1:01:59 on October 4, 1998 in Amsterdam, NL. (Time zone=+0100).

The HL7 Standard strongly recommends that all systems routinely send the time zone offset but does not require it. All HL7 systems are required to accept the time zone offset, but its implementation is application specific. For many applications the time of interest is the local time of the sender. For example, an application in the Eastern Standard Time zone receiving notification of an admission that takes place at 11:00 PM in San Francisco on December 11 would prefer to treat the admission as having occurred on December 11 rather than advancing the date to December 12.

Note:	The time zone [+/-ZZZZ], when used, is restricted to legally-defined time zones and is represented in HHMM format.

One exception to this rule would be a clinical system that processed patient data collected in a clinic and a nearby hospital that happens to be in a different time zone. Such applications may choose to convert the data to a common representation. Similar concerns apply to the transitions to and from daylight saving time. HL7 supports such requirements by requiring that the time zone information be present when the information is sent. It does not, however, specify which of the treatments discussed here will be applied by the receiving system.

2.9.48 TX - text data

String data meant for user display (on a terminal or printer). Such data would not necessarily be left justified since leading spaces may contribute greatly to the clarity of the presentation to the user. Because this type of data is intended for display, it may contain certain escape character sequences designed to control the display. Escape sequence formatting is defined later in this chapter in Section 2.10 “Use of escape sequences in text fields.” Leading spaces should be included. Trailing spaces should be removed. Example:

```
|  leading spaces are allowed.|
```

Since TX data is intended for display purposes, the repeat delimiter, when used with a TX data field, implies a series of repeating lines to be displayed on a printer or terminal. Therefore, the repeat delimiters are regarded as paragraph terminators or hard carriage returns (e.g., they would display as though a CR/LF were inserted in the text (DOS type system) or as though a LF were inserted into the text (UNIX style system)).

A receiving system would word-wrap the text between repeat delimiters in order to fit it into an arbitrarily sized display window but start any line beginning with a repeat delimiter on a new line.

Length: 65536

To include alternative character sets, use the appropriate escape sequence. See Section 2.16.9.18, "MSH-18 Character set (ID) 00692", and Section 2.16.9.20, "MSH-20 Alternate character set handling scheme (ID) 01317."

2.9.49 VH - visiting hours

```
Components:  <start day range (ID)> ^ <end day range (ID)> ^ <start hour range (TM)> ^ <end hour
             range (TM)>
```

Definition: This data type contains the hours when a patient location is open for visiting. Refer to *HL7 Table 0267 - Days of the week* for valid values for the first two components.

2.9.49.1 Start day range (ID)

Starting day of visiting hours range. See *HL7 Table 0267 - Days of the week* for valid values.

2.9.49.2 End day range (ID)

Ending day of visiting hours range. Starting day of visiting hours range. See *HL7 Table 0267 - Days of the week* for valid values

HL7 Table 0267 - Days of the week

Value	Description
SAT	Saturday
SUN	Sunday
MON	Monday
TUE	Tuesday
WED	Wednesday
THU	Thursday
FRI	Friday

2.9.49.3 Start hour range (TM)

Starting hour on starting day of visiting hours range (see first component, 2.9.49.1, “Start day range (ID)”).

2.9.49.4 End hour range (TM)

Ending hour on ending day of visiting hours range (see second component, 2.9.49.2, “End day range (ID)”).

2.9.50 VID – version identifier

```
Components:  <version ID (ID)> ^ <internationalization code (CE)> ^ <international version ID (CE)
```

2.9.50.1 Version ID (ID)

Used to identify the HL7 version. Refer to *HL7 Table 0104 – Version ID* for valid values.

2.9.50.2 Internationalization code (CE)

Used to identify the international affiliate country code. The values to be used are those of ISO 3166 - 1:1977. The ISO 3166 table has three separate forms of the country code: HL7 specifies that the 3-character (alphabetic) form be used for the country code.

Refer to *HL7 Table 0399 – Country code* for the 3-character codes as defined by ISO 3166 table.

2.9.50.3 International version ID (CE)

This field component identifies international affiliate’s version; it is especially important when the international affiliate has more than a single local version associated with a single US version.

2.9.51 XAD - extended address

```
Components: <street address (SAD)> ^ <other designation (ST)> ^ <city (ST)> ^ <state or province
            (ST)> ^ <zip or postal code (ST)> ^ <country (ID)> ^ < address type (ID)> ^ <other
            geographic designation (ST)> ^ <county/parish code (IS)> ^ <census tract (IS)> ^ <ad-
            dress representation code (ID)> ^ <address validity range (DR)>

Subcomponents of street address (SAD): <street or mailing address (ST)> & <street name (ST)> &
            <dwelling number (ST)>Subcomponents of address validity range (DR):  <date range
            start date/time (TS)> & <date range end date/time (TS)>
```

Note:	Replaces the AD data type as of v 2.3.

Length: 250

Example of usage for US:

```
|1234 Easy St.^Ste. 123^San Francisco^CA^95123^USA^B^^SF^|
```

This would be formatted for postal purposes as

```
1234 Easy St.
Ste. 123
San Francisco CA 95123
```

Example of usage for Australia:

```
|14th Floor^50 Paterson St^Coorparoo^QLD^4151|
```

This would be formatted for postal purposes using the same rules as for the American example as

14th Floor

50 Paterson St

Coorparoo QLD 4151

International note: Countries typically have a standard method of formatting addresses. This data type does not specify the formatting usages, only the components of a postal address.

2.9.51.1 Street address (SAD)

See section 2.9.38, "SAD – street address" for description of components.

2.9.51.2 Other designation (ST)

Second line of address. In US usage, it qualifies address. Examples: Suite 555 or Fourth Floor. When referencing an institution, this component specifies the street address.

2.9.51.3 City (ST)

This may be the name of the city, or district or place depending upon the national convention for formatting addresses for postal usage.

2.9.51.4 State or province (ST)

State or province should be represented by the official postal service codes for that country.

2.9.51.5 Zip or postal code (ST)

Zip or postal codes should be represented by the official codes for that country. In the US, the zip code takes the form 99999[-9999], while the Canadian postal code takes the form A9A9A9, and the Australian Postcode takes the form 9999

2.9.51.6 Country (ID)

Defines the country of the address. ISO 3166 provides a list of country codes that may be used. The ISO 3166 table has three separate forms of the country code: HL7 specifies that the 3-character (alphabetic) form be used for the country code. *HL7 Table 0399 – Country code* is defined to contain these 3-character codes.

2.9.51.7 Address type (ID)

Address type is optional and defined by *HL7 Table 0190 - Address type*.

2.9.51.8 Other geographic designation (ST)

Other geographic designation includes county, bioregion, SMSA, etc.

2.9.51.9 County/parish code (IS)

A code that represents the county in which the specified address resides. *User-defined Table 0289 - County/parish* is used as the HL7 identifier for the user-defined table of values for this component. When this component is used to represent the county (or parish), component 8 <other geographic designation> should not duplicate it (i.e., the use of <other geographic designation> to represent the county is allowed only for the purpose of backward compatibility, and should be discouraged in this and future versions of HL7).

Allowable values: codes defined by government.

User-defined Table 0289 – County/parish

Value	Description
	No suggested values defined

2.9.51.10 Census tract (IS)

A code that represents the census tract in which the specified address resides. *User-defined Table 0288 - Census tract* is used as the HL7 identifier for the user-defined table of values for this component.

Allowable Values: codes defined by government.

User-defined Table 0288 – Census tract

Value	Description
	No suggested values defined.

2.9.51.11 Address representation code (ID)

Different <name/address types> and representations of the same name/address should be described by repeating of this field, with different values of the <name/address type> and/or <name/address representation> component.

Note:	Also note that this new component remains in "alphabetic" representation with each repetition of the fields using these data types. I.e. even though the address may be represented in an ideographic character set, this component will remain represented in an alphabetic character set.

Refer to *HL7 table 0465 – Name/address representation* for valid values.

In general this component provides an indication of the representation provided by the data item. It does not necessarily specify the character sets used. Thus, even though the representation might provide an indication of what to expect, the sender is still free to encode the contents using whatever character set is desired. This component provides only hints for the receiver, so it can make choices regarding what it has been sent and what it is capable of displaying.

2.9.51.12 Address validity range (DR)

This component contains the start and end date/times which define the period in which this address was valid

2.9.51.12.1Date range start date/time (TS)

2.9.51.12.2Date range end date/time (TS)

2.9.52 XCN - extended composite ID number and name for persons

```
Components:  <ID number (ST)> ^ <family name (FN)> ^ <given name (ST)> ^ <second and further given
             names or initials thereof (ST)> ^ <suffix (e.g., JR or III) (ST)> ^ <prefix (e.g.,
             DR) (ST)> ^ <degree (e.g., MD) (IS)> ^ <source table (IS)> ^ <assigning authority
             (HD)> ^ <name type code (ID)>  ^ <identifier check digit (ST)> ^ <code identifying
             the check digit scheme employed (ID)> ^ <identifier type code (IS)> ^ <assigning fa-
             cility (HD)> ^ <name representation code (ID)> ^ <name context (CE)> ^ <name validity
             range (DR)> ^ <name assembly order (ID)>

Subcomponents of family name:  <surname (ST)> & <own surname prefix (ST)> & <own surname (ST)> &
             <surname prefix from partner/spouse (ST)> & <surname from partner/spouse (ST)>

Subcomponents of assigning authority:  <namespace ID (IS)> & <universal ID (ST)> & <universal ID
             type (ID)>
```

```
Subcomponents of assigning facility:  <namespace ID (IS)> & <universal ID (ST)> & <universal ID
            type (ID)>

Subcomponents of name context: <identifier (ST)> & <text (ST)> & <name of coding system (IS)> &
            <alternate identifier (ST)> & <alternate text (ST)> & <name of alternate coding sys-
            tem (IS)>

Subcomponents of name validity range:  <date range start date/time (TS)> & <date range end
            date/time (TS)>
```

Note: Replaces CN data type as of v 2.3.

Length: 250

This data type is used extensively appearing in the PV1, ORC, RXO, RXE, OBR and SCH segments , as well as others, where there is a need to specify the ID number and name of a person.

Example without assigning authority and assigning facility:

```
|1234567^Smith^John^J^III^DR^PHD^ADT01^^L^4^M11^MR|
```

Examples with assigning authority and assigning facility:

Dr. Samuel Semmelweiss's provider ID was assigned by the Provider Master and was first issued at Fairview Hospital within the University Hospitals System. Since IS table values (first component of the HD) were not used for assigning authority and assigning facility, components 2 and 3 of the HD data type are populated and demoted to sub-components as follows:

```
12188^Semmelweiss^Samuel^S^IV^Dr^MD^^&Provider Master.University
    Hospitals&L^L^9^M10^DN^&Fairview Hospital.University Hospitals&L^A
```

Ludwig van Beethoven's medical record number was assigned by the Master Patient Index and was first issued at Fairview Hospital within the University Hospitals System.

```
10535^van Beethoven&van^Ludwig^A^III^Dr^PHD^^&MPI.University
    Hospitals&L^L^3^M10^MR^&Fairview Hospital.University Hospitals&L^A
```

2.9.52.1 ID number (ST)

This string refers to the coded ID according to a user-defined table, defined by the 9th component. If the first component is present, either the source table or the assigning authority must be valued.

2.9.52.2 Family name (FN)

This component allows full specification of the surname of a person. Where appropriate, it differentiates the person's own surname from that of the person's partner or spouse, in cases where the person's name may contain elements from either name. It also permits messages to distinguish the surname prefix (such as "van" or "de") from the surname root. See section 2.9.19, “ FN - family name”.

2.9.52.3 Given name (ST)

First name.

2.9.52.4 Second and further given names or initials thereof (ST)

Multiple middle names may be included by separating them with spaces.

2.9.52.5 Suffix (ST)

Used to specify a name suffix (e.g., Jr. or III).

2.9.52.6 Prefix (ST)

Used to specify a name prefix (e.g., Dr.).

2.9.52.7 Degree (IS)

Used to specify an educational degree (e.g., MD). Refer to *User-defined Table 0360 – Degree* for suggested values.

2.9.52.8 Source table (IS)

User-defined Table 0297 – CN ID source is used as the HL7 identifier for the user-defined table of values for this component. Used to delineate the first component.

2.9.52.9 Assigning authority (HD)

The assigning authority is a unique identifier of the system (or organization or agency of department) that creates the data. *User-defined Table 0363 – Assigning authority* is used as the HL7 identifier for the user-defined table of values for the first sub-component of the HD component, <namespace ID>.

Note:	When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID* (referenced by the first sub-component of the HD component) may be re-defined (given a different user-defined table number and name) by the technical committee responsible for that segment. By site agreement, implementers may continue to use *User-defined Table 0300 – Namespace ID* for the first sub-component.

2.9.52.10 Name type code (ID)

A code that represents the type of name. Refer to *HL7 Table 0200 - Name type* for valid values (see Section 2.9.54.7, “Name type code (ID)”).

2.9.52.11 Identifier check digit (ST)

The check digit in this data type is not an add-on produced by the message processor. It is the check digit that is part of the identifying number used in the sending application. If the sending application does not include a self-generated check digit in the identifying number, this component should be valued null.

2.9.52.12 Code identifying the check digit scheme employed (ID)

Refer to *HL7 Table 0061 - Check digit scheme* for valid values.

2.9.52.13 Identifier type code (IS)

A code corresponding to the type of identifier. In some cases, this code may be used as a qualifier to the <assigning authority> component. Refer *to* *HL7 Table 0203 - Identifier type* for suggested values.

2.9.52.14 Assigning facility (HD)

The place or location identifier where the identifier was first assigned to the person. This component is not an inherent part of the identifier but rather part of the history of the identifier: as part of this data type, its existence is a convenience for certain intercommunicating systems.

Note:	When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID* (referenced by the first sub-component of the HD component) may be re-defined (given a different user-defined table number and name) by the technical committee responsible for that segment.

2.9.52.15 Name representation code (ID)

Different <name/address types> and representations of the same <name/address> should be described by repeating of this field, with different values of the <name/address type> and/or <name/address representation> component.

Note:	This new component remains in "alphabetic" representation with each repetition of the field using these data types. I.e.. even though the name may be represented in an ideographic character set, this component will remain represented in an alphabetic character set.

Refer to *HL7 Table 0465 – Name/address representation* for valid values.

In general this component provides an indication of the representation provided by the data item. It does not necessarily specify the character sets used. Thus, even though the representation might provide an indication of what to expect, the sender is still free to encode the contents using whatever character set is desired. This component provides only hints for the receiver, so it can make choices regarding what it has been sent and what it is capable of displaying.

2.9.52.16 Name context (CE)

This component is used to designate the context in which a name is used. The main use case is in Australian healthcare for indigenous patients who prefer to use different names when attending different healthcare institutions. Another use case occurs in the US where health practitioners can be licensed under slightly different names and the reporting of the correct name is vital for administrative purposes. Refer to *User-defined Table 0448 – Name context* for suggested values.

User-defined Table 0448 – Name context

Value	Description
	No suggested values defined

2.9.52.17 Name validity range (DR)

This component contains the start and end date/times that define the period during which this name was valid. See section 2.9.14, "DR - date/time range" for description of subcomponents.

2.9.52.18 Name assembly order (ID)

A code that represents the preferred display order of the components of this person name. *Refer to HL7 Table 0444 – Name assembly order* for valid values

2.9.53 XON - extended composite name and identification number for organizations

```
Components:  <organization name (ST)> ^ <organization name type code (IS)> ^ <ID number (NM)> ^
             <check digit (NM)> ^ <code identifying the check digit scheme employed (ID)> ^ <as-
             signing authority  (HD)> ^ <identifier type code (IS)> ^ <assigning facility ID (HD)>
             ^ <name representation code(ID)>

Subcomponents of assigning authority:  <namespace ID (IS)> & <universal ID (ST)> & <universal ID
             type (ID)>

Subcomponents of assigning facility:  <namespace ID (IS)> & <universal ID (ST)> & <universal ID
             type (ID)>
```

Length: 250

This data type is used in fields (e.g., PV2-23, NK1-13, PD1-3, OBR-44) to specify the name and ID number of an organization.

Example 1:

The ID for Fairview Hospital was assigned by the University Hospital enterprise's Hospital Master and was first issued at the Central Offices.

```
Fairview Hospital^L^716^9^M10^&Hospital Master.University
    Hositals&L^XX^&Central Offices.University Hospitals&L^A
```

Example 2:

Fairview Hospital has another ID that was issued by HCFA. Assigning Authority, HCFA, values only the first HD component, an IS data type and assigning facility is not relevant. This information might be transmitted accordingly:

```
Fairview Hospital^L^4544^3^M10^HCFA^XX^^A
```

2.9.53.1 Organization name (ST)

The name of the specified organization.

2.9.53.2 Organization name type code (IS)

A code that represents the type of name i.e., legal name, display name. Refer to *User-defined Table 0204 - Organizational name type* for suggested values.

User-defined Table 0204 - Organizational name type

Value	Description
A	Alias name
L	Legal name
D	Display name
SL	Stock exchange listing name

2.9.53.3 ID number (NM)

2.9.53.4 Check digit (NM)

The check digit in this data type is not an add-on produced by the message processor. It is the check digit that is part of the identifying number used in the sending application. If the sending application does not include a self-generated check digit in the identifying number, this component should be valued null.

2.9.53.5 Code identifying the check digit scheme employed (ID)

The check digit scheme codes are defined in *HL7 Table 0061 - Check digit scheme*.

2.9.53.6 Assigning authority (HD)

The assigning authority is a unique identifier of the system (or organization or agency or department) that creates the data. Assigning authorities are unique across a given HL7 implementation. *User-defined Table 0363 – Assigning authority* is used as the HL7 identifier for the user-defined table of values for the first sub-component of the HD component <namespace ID>.

Note:	When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID* (referenced by the first sub-component of the HD component) may be re-defined (given a different user-defined table number and name) by the technical committee responsible for that segment. By site agreement, implementors may continue to use *User-defined Table 0300 – Namespace ID* for the first sub-component.

2.9.53.7 Identifier type code (IS)

A code corresponding to the type of identifier. In some cases, this code may be used as a qualifier to the "Assigning authority" component. Refer to *HL7 Table 0203 - Identifier type* for suggested values.

2.9.53.8 Assigning facility ID (HD)

The place or location identifier where the identifier was first assigned to the person. This component is not an inherent part of the identifier but rather part of the history of the identifier: as part of this data type, its existence is a convenience for certain intercommunicating systems.

Note:	When the HD data type is used in a given segment as a component of a field of another data type, *User-defined Table 0300 - Namespace ID* (referenced by the first sub-component of the HD component) may be re-defined (given a different user-defined table number and name) by the technical committee responsible for that segment.

2.9.53.9 Name representation code (ID)

Different <name/address types> and representations of the same <name/address> should be described by repeating of this field, with different values of the <name/address type> and/or <name/address representation> component.

Note:	This new component remains in "alphabetic" representation with each repetition of the field using these data types, i.e. even though the name may be represented in an ideographic character set, this component will remain represented in an alphabetic character set.

Refer to *HL7 Table 0465 – Name/address representation code* for valid values.

In general this component provides an indication of the representation provided by the data item. It does not necessarily specify the character sets used. Thus, even though the representation might provide an indication of what to expect, the sender is still free to encode the contents using whatever character set is desired. This component provides only hints for the receiver, so it can make choices regarding what it has been sent and what it is capable of displaying.

2.9.54 XPN - extended person name

```
Components:    In Version 2.3, replaces the PN data type.  <family name (FN)> ^ <given name (ST)>
            ^ <second and further given names or initials thereof (ST)> ^ <suffix (e.g., JR or
            III) (ST)> ^ <prefix (e.g., DR) (ST)> ^ <degree (e.g., MD) (IS)> ^ <name type code
            (ID) > ^ <name representation code (ID)> ^ <name context (CE)> ^ <name validity range
            (DR)> ^ <name assembly order (ID)>

Subcomponents of family name:  <surname (ST)> ^ <own surname prefix (ST)> ^ <own surname (ST)> ^
            <surname prefix from partner/spouse (ST)> ^ <surname from partner/spouse (ST)>

Subcomponents of name context: <identifier (ST)> & <text (ST)> & <name of coding system (IS)> &
            <alternate identifier (ST)> & <alternate text (ST)> & <name of alternate coding sys-
            tem (IS)>

Subcomponents of name validity range:  <date range start date/time (TS)> & <date range end
            date/time (TS)>

Length: 250
```

Note:	Replaces PN data type as of v 2.3.

Example:

```
|Smith^John^J^III^DR^PHD^L|
```

2.9.54.1 Family name (FN)

This component allows full specification of the surname of a person. Where appropriate, it differentiates the person's own surname from that of the person's partner or spouse, in cases where the person's name may contain elements from either name. It also permits messages to distinguish the surname prefix (such as "van" or "de") from the surname root. See section 2.9.19, “FN - family name”.

2.9.54.2 Given name (ST)

First name.

2.9.54.3 Second and further given names or initials thereof (ST)

Multiple middle names may be included by separating them with spaces.

2.9.54.4 Suffix (ST)

Used to specify a name suffix (e.g., Jr. or III).

2.9.54.5 Prefix (ST)

Used to specify a name prefix (e.g., Dr.).

2.9.54.6 Degree (IS)

Used to specify an educational degree (e.g., MD). Refer to *User-defined Table 0360 – Degree* for suggested values.

User-defined Table 0360 - Degree

Value	Description
AAS	Associate of Applied Science
AA	Associate of Arts
ABA	Associate of Business Administration
AE	Associate of Engineering
AS	Associate of Science
BA	Bachelor of Arts
BBA	Bachelor of Business Administration
BE	Bachelor or Engineering
BFA	Bachelor of Fine Arts
BN	Bachelor of Nursing
BS	Bachelor of Science
BSL	Bachelor of Science – Law
BT	Bachelor of Theology
CER	Certificate
DIP	Diploma

Value	Description
DBA	Doctor of Business Administration
DED	Doctor of Education
PharmD	Doctor of Pharmacy
PHE	Doctor of Engineering
PHD	Doctor of Philosophy
PHS	Doctor of Science
MD	Doctor of Medicine
DO	Doctor of Osteopathy
HS	High School Graduate
JD	Juris Doctor
MA	Master of Arts
MBA	Master of Business Administration
MCE	Master of Civil Engineering
MDI	Master of Divinity
MED	Master of Education
MEE	Master of Electrical Engineering
ME	Master of Engineering
MFA	Master of Fine Arts
MME	Master of Mechanical Engineering
MS	Master of Science
MSL	Master of Science – Law
MT	Master of Theology
NG	Non-Graduate
SEC	Secretarial Certificate
TS	Trade School Graduate

2.9.54.7 Name type code (ID)

A code that represents the type of name. Refer to *HL7 Table 0200 - Name type* for valid values.

HL7 Table 0200 - Name type

Value	Description
A	Alias Name
B	Name at Birth
C	Adopted Name
D	Display Name
I	Licensing Name
L	Legal Name
M	Maiden Name

Value	Description
N	Nickname /"Call me" Name/Street Name
P	Name of Partner/Spouse (retained for backward compatibility only)
R	Registered Name (animals only)
S	Coded Pseudo-Name to ensure anonymity
T	Indigenous/Tribal/Community Name
U	Unspecified

Note: The content of Legal Name is country specific. In the US the legal name is the same as the current married name.

2.9.54.8 Name representation code (ID)

Different <name/address types> and representations of the same <name/address> should be described by repeating of this field, with different values of the <name/address type> and/or <name/address representation> component.

Note: This new component remains in "alphabetic" representation with each repetition of the field using these data types. I.e. even though the name may be represented in an ideographic character set, this component will remain represented in an alphabetic character set.

Refer to *HL7 Table 0465 – Name/address representation* for valid values.

In general this component provides an indication of the representation provided by the data item. It does not necessarily specify the character sets used. Thus, even though the representation might provide an indication of what to expect, the sender is still free to encode the contents using whatever character set is desired. This component provides only hints for the receiver, so it can make choices regarding what it has been sent and what it is capable of displaying.

2.9.54.9 Name context (CE)

```
Subcomponents of name context:  <identifier (ID)> & <text (ST)> & <name of coding system (IS)> &
            <alternate identifier (ID)> & <alternate text (ST)> & <name of alternate coding sys-
            tem (IS)>
```

This component is used to designate the context in which a name is used. The main use case is in Australian healthcare for indigenous patients who prefer to use different names when attending different healthcare institutions. Another use case occurs in the US where health practitioners can be licensed under slightly different names and the reporting of the correct name is vital for administrative purposes. Refer to *User-defined Table 0448 – Name context* for suggested values.

2.9.54.10 Name validity range (DR)

This component contains the start and end date/times which define the period during which this name was valid. See section 2.9.14, "DR - date/time range" for description of subcomponents.

2.9.54.11 Name assembly order (ID)

A code that represents the preferred display order of the components of this person name. Refer to *HL7 0444 – Name assembly order* for valid values.

2.9.55 XTN - extended telecommunication number

```
Components:  [NNN] [(999)]999-9999 [X99999] [B99999] [C any text] ^ <telecommunication use code
             (ID)> ^ <telecommunication equipment type (ID)> ^ <email address (ST)> ^ <country
```

```
code (NM)> ^ <area/city code (NM)> ^ <phone number (NM)> ^ <extension (NM)> ^ <any
text (ST)>
```

Note:	Replaces TN data type as of v 2.3

Length: 250

Example:

```
(415)555-3210^ORN^FX^
```

2.9.55.1 [(999)] 999-9999 [X99999] [C any text]

Defined as the TN data type (see Section 2.9.45, “TN - telephone number”), except that the length of the country access code has been increased to three.

2.9.55.2 Telecommunication use code (ID)

A code that represents a specific use of a telecommunication number. Refer *to* *HL7 Table 0201 - Telecommunication use code* for valid values.

HL7 Table 0201 - Telecommunication use code

Value	Description
PRN	Primary Residence Number
ORN	Other Residence Number
WPN	Work Number
VHN	Vacation Home Number
ASN	Answering Service Number
EMR	Emergency Number
NET	Network (email) Address
BPN	Beeper Number

2.9.55.3 Telecommunication equipment type (ID)

A code that represents the type of telecommunication equipment. Refer *to* *HL7 Table 0202 - Telecommunication equipment type* for valid values.

HL7 Table 0202 - Telecommunication equipment type

Value	Description
PH	Telephone
FX	Fax
MD	Modem
CP	Cellular Phone
BP	Beeper
Internet	Internet Address: Use Only If Telecommunication Use Code Is NET
X.400	X.400 email address: Use Only If Telecommunication Use Code Is NET

2.9.55.4 Email address (ST)

Internationalization note: To make this data type interoperate with CEN's Telecommunication data attribute group, we allow use of the second component for email addresses. The presence of an email address is specified by the addition of the value *NET* to the Phone Use Code table, and the type of Internet address is specified with the values *Internet* and *X.400* to the Phone Equipment Type table. When used for an Internet address, the first component of the XTN data type will be null. If the @-sign is being used as a subcomponent delimiter, the HL7 subcomponent escape sequence may be used when encoding an Internet address (see Section 2.10.1, "Formatting codes").

Note: Components five through nine reiterate the basic function of the first component in a delimited form that allows the expression of both local and international telephone numbers. In Version 2.3, the recommended form for the telephone number is to use the delimited form rather than the unstructured form supported by the first component (which is left in for backward compatibility only).

2.9.55.5 Country code (NM)

2.9.55.6 Area/city code (NM)

2.9.55.7 Phone number (NM)

2.9.55.8 Extension (NM)

2.9.55.9 Any text (ST)

2.10 USE OF ESCAPE SEQUENCES IN TEXT FIELDS

2.10.1 Formatting codes

When a field of type TX, FT, or CF is being encoded, the escape character may be used to signal certain special characteristics of portions of the text field. The escape character is whatever display ASCII character is specified in the <escape character> component of *MSH-2-encoding characters*. For purposes of this section, the character \ will be used to represent the character so designated in a message. An **escape sequence** consists of the escape character followed by an escape code ID of one character, zero (0) or more data characters, and another occurrence of the escape character. The following escape sequences are defined:

\H\	start highlighting
\N\	normal text (end highlighting)
\F\	field separator
\S\	component separator
\T\	subcomponent separator
\R\	repetition separator
\E\	escape character
\Xdddd...\	hexadecimal data
\Zdddd...\	locally defined escape sequence

The **escape sequences** for field separator, component separator, subcomponent separator, repetition separator, and escape character are also valid within an ST data field.

No escape sequence may contain a nested escape sequence.

2.10.2 Escape sequences supporting multiple character sets for, FT, ST, and TX data types

The following HL7 escape sequences are defined to support multiple character sets for fields, components and sub-components that are defined as data types FT, ST, and TX. They allow HL7 parsers to use escape codes (defined in the standards used below), without breaking, and without being non-conformant to the HL7 escape paradigm defined in this section.

\Cxxyy\ single-byte character set escape sequence with two hexadecimal values, xx and yy, that indicate the escape sequence defined for one of the character repertoires supported for the current message (i.e., ISO-IR xxx).

\Mxxyyzz\ multi-byte character set escape sequence with three hexadecimal values, xx, yy and zz. zz is optional.

Common character set escape sequences include the following which are defined in the standards mentioned:

Single-byte character sets:

\C2842\	ISO-IR6 G0 (ISO 646 : ASCII)
\C2D41\	ISO-IR100 (ISO 8859 : Latin Alphabet 1)
\C2D42\	ISO-IR101 (ISO 8859 : Latin Alphabet 2)
\C2D43\	ISO-IR109 (ISO 8859 : Latin Alphabet 3)
\C2D44\	ISO-IR110 (ISO 8859 : Latin Alphabet 4)
\C2D4C\	ISO-IR144 (ISO 8859 : Cyrillic)
\C2D47\	ISO-IR127 (ISO 8859 : Arabic)
\C2D46\	ISO-IR126 (ISO 8859 : Greek)
\C2D48\	ISO-IR138 (ISO 8859 : Hebrew)
\C2D4D\	ISO-IR148 (ISO 8859 : Latin Alphabet 5)
\C284A\	ISO-IR14 (JIS X 0201 -1976: Romaji)
\C2949\	ISO-IR13 (JIS X 0201 : Katakana)

Multi-byte codes:

\M2442\	ISO-IR87 (JIS X 0208 : Kanji, hiragana and katakana)
\M242844\	ISO-IR159 (JIS X 0212 : Supplementary Kanji)

2.10.3 Highlighting

In designating highlighting, the sending application is indicating that the characters that follow somehow should be made to stand out, but leaving the method of doing so to the receiving application. Depending on device characteristics and application style considerations, the receiving application may choose reverse video, boldface, underlining, blink, an alternate color or another means of highlighting the displayed data. For example the message fragment:

```
DSP|     TOTAL CHOLESTEROL     \H\240*\N\   [90 - 200]
```

might cause the following data to appear on a screen or report:

```
TOTAL CHOLESTEROL         240*    [90 - 200]
```

whereas another system may choose to show the 240* in red.

2.10.4 Special character

The special character escape sequences (\F\, \S\, \R\, \T\, and \E\) allow the corresponding characters to be included in the data in a text field, though the actual characters are reserved. For example, the message fragment

```
DSP|  TOTAL CHOLESTEROL     180  \F\90 - 200\F\
DSP|  \S\----------------\S\
```

would cause the following information to be displayed, given suitable assignment of separators:

```
TOTAL CHOLESTEROL     180  |90 - 200|
^----------------^
```

2.10.5 Hexadecimal

When the hexadecimal escape sequence (\Xdddd...\) is used the X should be followed by 1 or more pairs of hexadecimal digits (0, 1, . . . , 9, A, . . . , F). Consecutive pairs of the hexadecimal digits represent 8-bit binary values. The interpretation of the data is entirely left to an agreement between the sending and receiving applications that is beyond the scope of this Standard.

2.10.6 Formatted text

If the field is of the formatted text (FT) data type, formatting commands also may be surrounded by the escape character. Each command begins with the **.** (period) character. The following formatting commands are available:

.sp <number>	End current output line and skip <number> vertical spaces. <number> is a positive integer or absent. If <number> is absent, skip one space. The horizontal character position remains unchanged. Note that for purposes of compatibility with previous versions of HL7, "^\.sp\" is equivalent to "\.br\."
.br	Begin new output line. Set the horizontal position to the current left margin and increment the vertical position by 1.
.fi	Begin word wrap or fill mode. This is the default state. It can be changed to a no-wrap mode using the .nf command.
.nf	Begin no-wrap mode.

.in <number>	Indent <number> of spaces, where <number> is a positive or negative integer. This command cannot appear after the first printable character of a line.
.ti <number>	Temporarily indent <number> of spaces where number is a positive or negative integer. This command cannot appear after the first printable character of a line.
.sk < number>	Skip <number> spaces to the right.
.ce	End current output line and center the next line.

The component separator that marks each line defines the extent of the temporary indent command (.ti), and the beginning of each line in the no-wrap mode (.nf). Examples of formatting instructions that are NOT included in this data type include: width of display, position on page or screen, and type of output devices.

Figure 2-3 is an example of the FT data type from a radiology impression section of a radiology report:

Figure 2-3. Formatted text as transmitted

```
\.in+4\\.ti-4\ 1.  The cardiomediastinal silhouette is now within normal limits.^\.sp\\.ti-4\
2.   Lung fields show minimal ground glass appearance.^\.sp\\.ti-4\ 3.  A loop of colon
visible in the left upper quadrant is distinctly abnormal with the appearance of mucosal
effacement suggesting colitis.\.in-4\|
```

Figure 2-4 shows one way of presenting the data in Figure 2-3. The receiving system can create many other interpretations by varying the right margin.

Figure 2-4. Formatted text in one possible presentation

```
1.  The cardiomediastinal silhouette is now within normal limits.

2.  Lung fields show minimal ground glass appearance.

3.  A loop of colon visible in the left upper quadrant is distinctly abnormal with the
    appearance of mucosal effacement suggesting colitis.
```

2.10.7 Local

When the local escape sequence (\Zdddd...\) is used the Z should be followed by characters that are valid in a TX field. The interpretation of the data is entirely left to an agreement between the sending and receiving applications that is beyond the scope of this Standard.

2.11 MESSAGE CONSTRUCTION RULES

Note: These message construction rules define the standard HL7 encoding rules, creating variable length delimited messages. Although only one set of encoding rules is defined as a standard in HL7 Version 2.3, other encoding rules are possible (but since they are non-standard, they may only be used by a site-specific agreement).

Step 1 Construct the segments in the order defined for the message. Each message is constructed as follows:

a) the first three characters are the segment ID code

b) each data field in sequence is inserted in the segment in the following manner:

1) a field separator is placed in the segment

2) if the value is not present, no further characters are required

3) if the value is present, but null, the characters "" (two consecutive double quotation marks) are placed in the field

4) otherwise, place the characters of the value in the segment. As many characters can be included as the maximum defined for the data field. It is not necessary, and is undesirable, to pad fields to fixed lengths. Padding to fixed lengths is permitted. Encode the individual data fields as shown in Section 2.9, "Data types."

5) if the field definition calls for a field to be broken into components, the following rules are used:

   i. if more than one component is included they are separated by the component separator

   ii. components that are present but null are represented by the characters ""

   iii. components that are not present are treated by including no characters in the component

   iv. components that are not present at the end of a field need not be represented by component separators. For example, the two data fields are equivalent:

   |ABC^DEF^^| and |ABC^DEF|.

6) if the component definition calls for a component to be broken into subcomponents, the following rules are used:

   i. if more than one subcomponent is included they are separated by the subcomponent separator

   ii. subcomponents that are present but null are represented by the characters ""

   iii. subcomponents that are not present are treated by including no characters in the subcomponent

   iv. subcomponents that are not present at the end of a component need not be represented by subcomponent separators. For example, the two data components are equivalent:

   ^XXX&YYY&&^ and ^XXX&YYY^.

7) if the field definition permits repetition of a field, the repetition separator is used only if more than one occurrence is transmitted. In such a case, the repetition separator is placed between occurrences. If three occurrences are transmitted, two repetition separators are used.)

   In the example below, two occurrences of telephone number are being sent:

   |234-7120~599-1288B1234|

c) repeat Step 1b while there are any fields present to be sent. If all the data fields remaining in the segment definition are not present there is no requirement to include any more delimiters.

d) end each segment with an ASCII carriage return character

Step 2 Repeat Step 1 until all segments have been generated.

The following rules apply to receiving HL7 messages and converting their contents to data values:

a) ignore segments, fields, components, subcomponents, and extra repetitions of a field that are present but were not expected

b) treat segments that were expected but are not present as consisting entirely of fields that are not present

c) treat fields and components that are expected but were not included in a segment as not present.

2.11.1 Encoding rules notes

If a segment is to be continued across messages, use the extended encoding rules. These rules are defined in terms of the more general message continuation protocol (see Section 2.15.2, "Continuation messages and segments").

2.11.2 Version compatibility definition

The above rules for receiving HL7 messages and converting their contents to data values allow the following definition of a backward compatibility requirement between the 2.x versions of HL7:

a) New messages may be introduced.

b) New segments may be introduced to an existing message. In general these will be introduced at the end of a message, but they may be introduced elsewhere within the message if the segment hierarchy makes this necessary.

c) New fields may be added at the end of a segment, new components may be added at the end of a field, new subcomponents may be added at the end of a component, and a non-repeating field may be made repeating.

d) Existing optional segments, fields, components, and subcomponents may be made conditional or required.

If a non-repeating field is made repeating, the first instance of that repeating field must have the same meaning as the non-repeating field had in the prior version of HL7.

For existing fields in existing segments, data types may be changed by the above rule in point "c" if the leftmost (prior version) part of the field has the same meaning as it had in the prior version of HL7. In other words, if the new parts of the field (those that are part of the new data type) are ignored, what remains is the old field (defined by the old data type), which has the same meaning as it had in the prior version of HL7. When optional elements are made required or conditional, what remains for older versions is not changed.

2.12 CHAPTER FORMATS FOR DEFINING HL7 MESSAGES

Subsequent chapters of this document describe messages that are exchanged among applications in functionally-specific situations. Each chapter is organized as follows:

a) **purpose**. This is an overview describing the purpose of the chapter, general information and concepts.

b) **trigger events and messages**. There is a list of the trigger events. For each trigger event the messages that are exchanged when the trigger event occurs are defined using the HL7 abstract message syntax as follows:

Each message is defined in special notation that lists the segment IDs in the order they would appear in the message. Braces, { . . . }, indicate one or more repetitions of the enclosed group of segments. (Of course, the group may contain only a single segment.) Brackets, [. . .], show that the enclosed group of segments is optional. If a group of segments is optional and may repeat it should be enclosed in brackets and braces, { [. . .] }.

Note:	[{...}] and {[...]} are equivalent.

Whenever braces or brackets enclose more than one segment ID a special stylistic convention is used to help the reader understand the hierarchy of repetition. For example, the first segment ID appears on the same line as the brace, two columns to the right. The subsequent segment IDs appear under the first. The closing brace appears on a line of its own in the same column as the opening brace. This convention is an optional convenience to the user. If there is conflict between its use and the braces that appear in a message schematic, the braces define the actual grouping of segments that is permitted.

A choice of one segment from a group of segments is indicated by using angle brackets to delimit the group and vertical bar delimiters between the several segments.

Example: The ORM^O01, as described in chapter 4 section 4.4.1, allows a choice of order detail segments. The choice would be represented as follows:

<OBR | RQD | RQ1 | RXO | ODS | ODT>

c) **message segments**. The segments defined in a chapter are then listed in a functional order designed to maximize conceptual clarity.

d) **examples**. Complete messages are included.

e) **implementation considerations**. Special supplementary information is presented here. This includes issues that must be addressed in planning an implementation.

f) **outstanding issues**. Issues still under consideration or requiring consideration are listed here.

Consider the hypothetical triggering event **a widget report is requested**. It might be served by the Widget Request (WRQ) and Widget Report (WRP) messages. These would be defined in the Widget chapter (say Chapter XX). The Widget Request message might consist of the following segments: Message Header (MSH), Widget ID (WID). The Widget Report message might consist of the following segments: Message Header (MSH), Message acknowledgment (MSA), one or more Widget Description (WDN) Segments each of which is followed by a single Widget Portion segment (WPN) followed by zero or more Widget Portion Detail (WPD) segments.

2.12.1 HL7 abstract message syntax example

The schematic form for this hypothetical exchange of messages is shown in Figure 2-5:

Figure 2-5. Hypothetical schematic message

Trigger Event: WIDGET REPORT IS REQUESTED

WRQ	Widget Request	Chapter
MSH	Message Header	2
WID	Widget ID	XX

WRP	Widget Report	Chapter
MSH	Message Header	2
MSA	Message Acknowledgment	2
{ WDN	Widget Description	XX
WPN	Widget Portion	XX
}		

The WID, WDN, WPN, and WPD segments would be defined by the widget committee in the widget chapter, as designated by the Arabic numeral XX in the right column. The MSH and MSA segments, although included in the widget messages, are defined in another chapter. They are incorporated by reference into the widget chapter by the chapter number XX.

On the other hand, the widget committee might decide that the WPN and WPD segments should appear in pairs, but the pairs are optional and can repeat. Then the schematic for the WRP message would be as shown in Figure 2-6.

Figure 2-6. WPN and WPD segments in pairs

WRF	Widget Report	Chapter
MSH	Message Header	2
MSA	Message Acknowledgment	2
{ WDN	Widget Description	XX
[{ WPN	Widget Portion	XX
WPD	Widget Portion Detail	XX
}]		
}		

If the widget committee determined that at least one pair of WPN and WPD segments must follow a WDN, then the notation would be as shown in Figure 2-7.

Figure 2-7. At least one pair of WPN and WPD

WRP	Widget Report	Chapter
MSH	Message Header	2
MSA	Message Acknowledgment	2
{ WDN	Widget Description	XX
{ WPN	Widget Portion	XX
WPD	Widget Portion Detail	XX
}		
}		

2.13 APPLICATION (LEVEL 7) PROCESSING RULES

2.13.1 Original and enhanced processing rules

The processing rules described here apply to all exchanges of messages, whether or not the HL7 encoding rules or Lower Layer Protocols are used. They represent the primary message processing mode. Certain variants are documented in Section 2.13.2, “Application (level 7) processing rules, deferred processing two phase reply (original acknowledgment mode only).” These include:

a) the application processing rules for a special processing mode, deferred processing. This mode remains in the specification only for backward compatibility.

b) an optional sequence number protocol

c) an optional protocol for continuing a very long message

The processing rules were extended in Version 2.2 of the Standard. The extensions provide a greater degree of flexibility in the way that messages can be acknowledged, as specified by several new fields in the Message Header segment. To provide backward compatibility with prior versions, the absence of these fields implies that the extended processing rules are not used. In the remainder of this section the extended mode is called the enhanced acknowledgment mode; the prior version is called the original acknowledgment mode.

Because the protocol describes an exchange of messages, it is described in terms of two entities, the initiating and responding systems. Each is both a sender and receiver of messages. The initiating system sends first and then receives, while the responding system receives and then sends.

In overview this exchange proceeds as follows:

Step 1 the initiating system constructs an HL7 message from application data and sends it to the responding system

Step 2 responder receives message and

2.1 when the original acknowledgment rules apply:

a) validates the message syntactically and against the detailed rules described in Section 2.13.1.2.1, If it fails, a reject message is constructed by the protocol software and returned to the initiator; if it does not fail, continue to the next step (2.1,b)

b) passes the message to the application, which:

1) creates a response message, or

2) creates an error message, or

3) creates a reject message

c) sends the response, error, or reject message

Initiator passes the message to the initiating application.

2.2 when enhanced acknowledgment rules apply:

See section 2.13.1.2.2

a) the responding system receives the message and commits it to safe storage. This means that the responding system accepts the responsibility for the message in a manner that releases the sending system from any obligation to resend the message. The responding system now checks the message header record to determine whether or not the initiating system requires an accept acknowledgment message indicating successful receipt and secure storage of the message. If it does, the accept acknowledgment message is constructed and returned to the initiator.

b) at this point, the requirements of the applications involved in the interface determine whether or not more information needs to be exchanged. This exchange is referred to as an application acknowledgment and includes information ranging from simple validation to a complex application-dependent response. If the receiving system is expected to return application-dependent information, it initiates another exchange when this information is available. This time, the roles of initiator and responder are reversed.

The details follow.

2.13.1.1 Initiation

The initiating application creates a message with data values as defined in the appropriate chapter of this Standard. The fields shown below should be valued in the MSH segment (as defined under the MSH segment definition of this chapter). The message is encoded according to the applicable rules and sent to the lower level protocols, which will attempt to deliver it to the responding application. (For definitions of the MSH fields see Section 2.16.9, “MSH - message header segment”)

Field	Notes
MSH-3-sending application	
MSH-4-sending facility	
MSH-5-receiving application	
MSH-6-receiving facility	
MSH-7-date/time of message	
MSH-9-message type	
MSH-10-message control ID	Unique identifier used to relate the response to the initial message.
MSH-11-processing ID	
MSH-12-version ID	
MSH-13-sequence number	
MSH-14-continuation pointer	Used in implementation of message continuation protocol. See Section 2.15.2, “Continuation messages and segments”. Also see chapter 5.

Certain other fields in the MSH segment are required for the operation of the HL7 encoding rules; they will not be relevant if other encoding rules are employed.

The event code in the second component of *MSH-9-message type* is redundantly shown elsewhere in some messages. For example, the same information is in the EVN segment of the ADT message. This is for

compatibility with prior versions of the HL7 protocol. Newly-defined messages should only show the event code in *MSH-9-message type*.

2.13.1.2 Response

The protocol software in the responding system does one of the following:

2.13.1.2.1 When the original acknowledgment rules apply

Note:	Both MSH-15-accept acknowledgment type and MSH-16-application acknowledgment type are null or not present.

a) accepts the message

b) validates it against at least the following criteria:

1) the value in *MSH-9-message type* is one that is acceptable to the receiver

2) the value in *MSH-12-version ID* is acceptable to the receiver

3) the value in *MSH-11-processing ID* is appropriate for the application process handling the message

If any of these edits fail, the protocol software rejects the message. That is, it creates an ACK message with **AR** in *MSA-1-acknowledgment code*.

c) if the message passes the edits, the message is passed to the receiving application, which performs one of these functions:

1) process the message successfully, generating the functional response message with a value of **AA** in *MSA-1-acknowledgment code*.

-OR-

2) send an error response, providing error information in functional segments to be included in the response message with a value of **AE** in *MSA-1-acknowledgment code*.

-OR-

3) fail to process (reject) the message for reasons unrelated to its content or format (system down, internal error, etc.). For most such problems it is likely that the responding system will be able to accept the same message at a later time. The implementers must decide on an application-specific basis whether the message should be automatically sent again. The response message contains a value of **AR** in *MSA-1-acknowledgment code*.

d) passes the message to the initiating system

e) the protocol software in the initiating system passes the response message to the initiating application

In all the responses described above the following values are put in the MSA segment. Note that the field definitions for the MSA segment fields are in Section 2.16.8, “MSA - message acknowledgment segment”:

Field	Notes
MSA-1-acknowledgment code	As described above.
MSA-2-message control ID	MSH-10-message control ID from MSH segment of incoming message.
MSA-3-text message	Text description of error.
MSA-4-expected sequence number	As described in Section 2.15.1, “Sequence number protocol,” (if the sequence number protocol is being used).
MSA-5-delayed acknowledgment	For use only as described in Section Application (level

type	7) processing rules, deferred processing two phase reply (original acknowledgment **mode only)**."

The MSH segment in the response is constructed anew following the rules used to create the initial message described above. In particular, *MSH-7-date/time of message* and *MSH-10-message control ID* refer to the response message; they are not echoes of the fields in the initial message. *MSH-5-receiving application*, *MSH-6-receiving facility*, and *MSH-11-processing ID* contain codes that are copied from *MSH-3-sending application*, *MSH-4-sending facility* and *MSH-11-processing ID* in the initiating message.

2.13.1.2.2 When enhanced acknowledgment rules apply

Note:	At least one of MSH-15-accept acknowledgment type or MSH-16-application acknowledgment type is not null.

a) accepts the message

b) makes an initial determination as to whether or not the message can be accepted, based on factors such as:

1) the status of the interface

2) the availability of safe storage onto which the message can be saved

3) the syntactical correctness of the message, if the design of the receiving system includes this type of validation at this phase

4) the values of *MSH-9-message type*, *MSH-12-version ID*, and *MSH-11-processing ID*, if the design of the receiving system includes this type of validation at this phase

c) examines the Message Header segment (MSH) to determine whether or not the initiating system requires an accept acknowledgment.

If it does, the responding system returns a general acknowledgment message (ACK) with:

1) a commit accept (CA) in *MSA-1-acknowledgment code* if the message can be accepted for processing

2) a commit reject (CR) in *MSA-1-acknowledgment code* if the one of the values of *MSH-9-message type*, *MSH-12-version ID* or *MSH-11-processing ID* is not acceptable to the receiving application

3) a commit error (CE) in *MSA-1-acknowledgment code* if the message cannot be accepted for any other reason (e.g., sequence number error)

For this response, the following values are put in the MSA segment. Note that the field definitions for the MSA segment fields are in Section 2.16.8, "MSA - message acknowledgment segment":

Field	Notes
MSA-2-message control ID	MSH-10-message control ID from the incoming message.
MSA-1-acknowledgment code	As described above.
MSA-3-text message	Text description of error.
MSA-4-expected sequence number	As described in Section 2.15.1, "Sequence number protocol" (if the sequence number protocol is being used).

The MSH segment in the response is constructed anew following the rules used to create the initial message described above. In particular, *MSH-7-date/time of message* and *MSH-10-message control ID* refer to the response message; they are not echoes of the fields in the initial message. *MSH-5-receiving application*, *MSH-6-receiving facility*, and *MSH-11-processing ID* contain codes that are copied from *MSH-3-sending application*, *MSH-4-sending facility* and *MSH-11-processing ID* in the initiating message.

Note:	MSH-15-accept acknowledgment type and MSH-16-application acknowledgment type are not valued (not present or null). At this point, the accept portion of this message exchange is considered complete.

d) If the message header segment indicates that the initiating system also requires an application acknowledgment, this will be returned as the initial message of a later exchange.

For this response, the following values are put in the MSA segment. Note that the field definitions for the MSA segment fields are in Section 2.16.8, "MSA - message acknowledgment segment":

Field	Notes
MSA-2-message control ID	Identifies the initial message from the original initiating system as defined in Section 2.13.1.1, "Initiation".
MSA-1-acknowledgment code	Uses the application (processing) acknowledgment codes as described in Section 2.13.1.2.1, "When the original acknowledgment rules apply"
MSA-3-text message	Text description of error.

For this message, the receiving system acts as the initiator. Since the message it sends is application-specific, the layouts of these application-level response messages are defined in the relevant application-specific chapter. If needed, this application acknowledgment message can itself require (in *MSH-15-accept acknowledgment type*) an accept acknowledgment message (MSA). *MSH-16-application acknowledgment type*, however, is always null, since the protocol does not allow the application acknowledgment message to have an application acknowledgment.

At this point, the application acknowledgment portion of this message exchange is considered complete.

If the processing on the receiving system goes through multiple stages, chapter-defined messages may be used to relay status or informational changes to other systems (including the original initiating system). Such messages are not part of the acknowledgment scheme for the original message, but are considered to be independent messages triggered by events on the (original) responding system.

Note:	The original acknowledgment protocol is equivalent to the enhanced acknowledgment protocol with *MSH-15-accept acknowledgment type* = NE and *MSH-16-application acknowledgment type* = AL, and with the application acknowledgment message defined so that it never requires an accept acknowledgment (*MSH-15-accept acknowledgment type* = NE).

2.13.2 Application (level 7) processing rules, deferred processing two phase reply (original acknowledgment mode only)

(This section remains in the specification only for reasons of providing backward compatibility: it is to be used only with the original acknowledgment protocol. For the original acknowledgment protocol, it creates a generic form of an asynchronous application level acknowledgment, the MCF message.)

The application processing rules for deferred processing are described here. In this mode the responding system sends an acknowledgment to the initiating system that means the message has been placed in some type of secure environment (e.g., disk storage), and the receiving system commits to processing it within a reasonable amount of time, if a) the message contains the necessary information, and b) nothing causes the message's request for action to be canceled before the responding system processes the request.

Note:	Neither of these two conditions is completely checked at the time of the first acknowledgment. They are both checked at the time of processing.

The receipt of the first delayed acknowledgment by the initiating system means that the responding system has taken responsibility for the subsequent processing of the message. This also implies that the initiating system no longer needs to keep the particular message in its current form to send out later. For example, if

the sending system were maintaining a queue of outgoing messages, the particular message could be deleted from the output queue at this point.

The receipt of the second delayed acknowledgment message informs the initiating application of either: a) the application's successful processing of the initial message, or b) an error that prevented its processing. If the receiving application needs to return detailed change of status information, an application-specific message will be used. An example of the latter is the General Order message (ORM) described in Chapter 4.

The general delayed acknowledgment protocol is implemented on a site-specific and application-specific basis as needed. At a particular site, for a given transaction type the choices are:

a) do not allow deferred acknowledgments

b) all messages will have a deferred acknowledgment

c) only exceptional cases (errors) will receive the deferred acknowledgment

In overview the processing for options b) and c) proceeds as follows:

Initiator receives message from sending application and sends it to the responding system.

The responding system receives the message from the initiating system and

a) partially validates it syntactically and against the detailed rules described in Section 2.13.1, "Original and enhanced processing rules." This validation need not be complete but should be sufficient to determine the application that will ultimately respond to the message. If this validation fails, a reject message is constructed by the protocol software and returned to the initiator.

b) (if the message passes this validation) stores it and constructs a response message that simply acknowledges receipt. *MSA-5-delayed acknowledgment type* then has a value of **D**.

c) subsequently passes the message to the application, which:

   1) creates a response message, or

   2) creates an error message, or

   3) creates a reject message

d) The protocol software sends the response, error, or reject message to the initiating system as an unsolicited update with no value present in *MSA-5-delayed acknowledgment type*.

The protocol software of the initiating system responds to the response, error, or reject message with simple acknowledgment and passes it to the initiating application.

The details follow.

2.13.2.1 Initiation

The rules for creating the initial message are exactly as defined in Section 2.13.1, "Original and enhanced processing rules," for the original acknowledgment rules.

2.13.2.2 Response

The processing in the responding system follows this pattern:

a) the protocol software accepts the message and validates it against at least the following criteria:

1) the value in *MSH-9-message type* is one that is acceptable to the receiver

2) the value in *MSH-12-version ID* is acceptable to the receiver

3) the value in *MSH-11-processing ID* is appropriate for the application process handling the message

If any of these edits fail, the protocol software rejects the message. That is, it creates an ACK message with **AR** in *MSA-1-acknowledgment code*.

b) If the message passes the edits, the protocol software stores it and generates a response message of type ACK with a value of **AA** in *MSA-1-acknowledgment code* and **D** in *MSA-5-delayed acknowledgment type*.

c) Subsequently the protocol software passes the message to the application, which performs one of these functions:

1) processes the message successfully, generating the functional response message (message type MCF) with a value of **AA** in *MSA-1-acknowledgment code*.

- OR -

2) creates an error response, providing error information in functional segments to be included in the response message, which has a value of **AE** in *MSA-1-acknowledgment code*.

- OR -

3) fails to process (rejects) the message for reasons unrelated to its content or format (system down, internal error, etc.) For most such problems it is likely that the responding system will be able to accept the same message at a later time. The implementors must decide on an application-specific basis whether the message should be automatically sent again. The MSA segment of the response message contains a value of **AR** in *MSA-1-acknowledgment code*.

d) the application passes the message to the protocol software, which constructs a message of type MCF with **F** in *MSA-5-delayed acknowledgment type*.

e) the protocol software passes the message to the initiating system as an unsolicited update.

f) the protocol software in the initiating system passes the response message to the initiating application and generates a simple ACK message. No value is present in *MSA-5-delayed acknowledgment type*.

All other values are put in the MSA segment as described in Section 2.13.1, “Original and enhanced processing rules.”

2.14 ACKNOWLEDGMENT MESSAGES

Acknowledgment messages may be defined on an application basis. However the simple general acknowledgment message (ACK) may be used where the application does not define a special message (application level acknowledgment) and in other cases as described in Section 2.13.1, “Original and enhanced processing rules.” ***The MCF message is included only for backward compatibility with HL7 Version 2.1 (see Section 2.12.2, “***Application (level 7) processing rules, deferred processing two phase reply (original acknowledgment mode only)***”).***

2.14.1 ACK - general acknowledgment

The simple general acknowledgment (ACK) can be used where the application does not define a special application level acknowledgment message or where there has been an error that precludes application

processing. It is also used for accept level acknowledgments. The details are described in Section 2.13.1, "Original and enhanced processing rules."

ACK^varies^ACK	General Acknowledgment	Chapter
MSH	Message Header	2
MSA	Message Acknowledgment	2
[ERR]	Error	2

Note: For the general acknowledgment (ACK) message, the value of *MSH-9-2-Trigger event* is equal to the value of *MSH-9-2-Trigger event* in the query message being acknowledged. The value of *MSH-9-3-Message structure* for the general acknowledgment message is always ACK.

2.14.2 MCF - delayed acknowledgment

This message remains in the specification only for reasons of backward compatibility with HL7 Version 2.1. It is used as part of the protocol which creates a generic form of an asynchronous application level acknowledgment, the MCF message. See Section 2.13.2.2, "Response."

The first MCF message, sent after the initial receipt has the following structure.

MCF^varies^ACK	Delayed Acknowledgment	Chapter
MSH	Message Header	2
MSA	Message Acknowledgment	2
[ERR]	Error	2

The second MCF message, sent after application processing, has this structure:

MCF^varies^ACK	Delayed Acknowledgment	Chapter
MSH	Message Header	2
MSA	Message Acknowledgment	2
[ERR]	Error	2

2.15 SPECIAL HL7 PROTOCOLS

This section contains several extensions to the basic HL7 message protocol. These extensions represent implementation choices, and are to be used on a site-specific and application-specific basis as needed.

2.15.1 Sequence number protocol

For certain types of data transactions between systems the issue of keeping databases synchronized is critical. An example is an ancillary system such as lab, which needs to know the locations of all inpatients to route stat results correctly. If the lab receives an ADT transaction out of sequence, the census/location information may be incorrect. Although it is true that a simple one-to-one acknowledgment scheme can prevent out-of-sequence transactions between any two systems, only the use of sequence numbers can prevent duplicate transactions.

Note: Although this sequence number protocol is limited to the use of sequence numbers on a single transaction stream between two applications, this sequencing protocol is sufficiently robust to allow the design of HL7-compatible store-and-forward applications.

a) initial conditions:

1) the system receiving the data stream is expected to store the sequence number of the most recently accepted transaction in a secure fashion before acknowledging that transaction. This stored sequence number allows comparison with the next transaction's sequence number, and the implementation of fault-tolerant restart capabilities.

2) the initiating system keeps a queue of outgoing transactions indexed by the sequence number. The length of this queue must be negotiated as part of the design process for a given link. The minimum length for this queue is one.

3) the sequence number is a positive (non-zero) integer; and it is incremented by one (by the initiating system) for each successive transaction.

b) starting the link:

1) the value of 0 (zero) for a sequence number is reserved: it is allowed only when the initiating system (re-)starts the link.

2) if the receiving system gets a transaction with a 0 (zero) in the sequence number field, it should respond with a general acknowledgment message whose MSA contains a sequence number one greater than the sequence number of the last transaction it accepted in the Expected Sequence Number field. If this value does not exist (as on the first startup of a given link), the MSA should contain a sequence number of -1, meaning that the receiving system will use the positive, non-zero sequence number of the first transaction it accepts as its initial sequence number (see resynching the link, item **e** below).

3) the initiating system then sends the transaction indexed by the expected sequence number (if that expected transaction is still on its queue). Otherwise the link is frozen until an operator intervenes.

c) normal operation of the link:

As it accepts each transaction, the receiving system securely stores the sequence number (which agrees with its expected sequence number), and then acknowledges the message by echoing the sequence number in *MSA-4-expected sequence number*.

d) error conditions (from point of view of initiating system). These are generated by the receiving system, by its comparison of the sequence number sent out (with the MSH in *MSH-13-sequence number*) with the expected sequence number (*MSA-4-expected sequence number* received with the MSA).

1) **expected sequence number is one greater than current value**. The previous acknowledgment was lost. That transaction was sent again. Correct by sending next transaction.

2) **expected sequence number less than current value**. Initiating system can try starting again by issuing a transaction with a sequence number of zero; or freeze the link for operator intervention.

3) **other errors**: freeze the link for operator intervention

e) forcing resynchronization of sequence numbers across the link. The value of -1 for a sequence number is reserved: it is allowed only when the initiating system is resynching the link. Thus if the receiving system gets a value of -1 in the sequence number field, it should return a general acknowledgment message with a -1 in the expected sequence number field. The receiving system then resets its sequence number, using the non-zero positive sequence number of the next transaction it accepts.

f) notes

When the initiating system sends a message with a sequence number of **0** or **-1** (see b or e above), the segments beyond the MSH need not be present in the message, or, if present, all fields can be null. In terms of the responding system, for these two cases, only a General acknowledgment message is needed.

2.15.2 Continuation messages and segments

Sometimes, implementation limitations require that large messages or segments be broken into manageable chunks. We use the term “fragmentation” to describe how a logical message is broken into one or more separate HL7 messages. HL7 consciously identifies two situations where this may happen.

- First, a single segment may be too large. HL7 uses the “ADD” segment to handle breaking a single segment into several smaller segments.

- Second, a single HL7 message may be too large. HL7 uses the DSC segment and the continuation protocol to handle message fragmentation.

Note:	HL7 does not define what "too large" means. Acceptable values are subject to site negotiations.

See chapter 5 for a discussion of the continuation pointer segment and the continuation pointer field, and their use in the continuation of responses to queries and in the continuation of unsolicited update messages.

2.15.2.1 Segment fragmentation/continuation using the ADD segment

Beginning with version 2.4, the ADD segment can be used within a message to break a long segment into shorter segments within a single HL7 message.

Note:	Unless some explicit agreement exists between systems, a receiving application should not infer semantic meaning from the placement of the ADD segment.

To break a large segment,

a) the segment being continued (call it ANY for this example) is ended at an arbitrary character position and terminated with the standard segment terminator (carriage return).

b) the following segment is the ADD segment. All characters after the ADD and field separator ("|") are logically part of the preceding segment. All succeeding consecutive ADD segments contribute characters to the ANY segment until a non ADD segment is found.

c) An ADD segment with no field separator takes on special meaning. See Section 2.14.2.3, "Segment fragmentation across messages."

For example, segment "C" can be fragmented within an HL7 message as follows

```
A|1
B|2
C|34
ADD|5|678|
ADD|90
D|1
```

This is logically the same as

```
A|1
B|2
C|345|678|90
D|1
```

Note that the "|" at the end of the first ADD segment is part of the value, while the first "|" of each ADD is not.

2.15.2.2 Segment fragmentation/continuation using the DSC segment

When a message itself must be fragmented and sent as several HL7 messages, the DSC segment is used.

a) First, the logical message is broken after an arbitrary segment.

b) Next, a DSC segment is sent. The *DSC-1-Continuation pointer* field will contain a unique value which is used to match a subsequent message with this specific value.

c) The DSC terminates the first fragment of the logical message.

d) A subsequent message will contain in *MSH-14-Continuation pointer*, a value which matches the value from DSC-1. (The presence of a value in MSH-14 indicates that the message is a fragment of an earlier message.). Each subsequent message will have its own unique value for *MSH-10-Message control ID*. Coordination between *DSC-1-Continuation pointer* and the subsequent message's *MSH-14-Continuation pointer* is used to link the fragments in their proper order.

e) The logical message is the concatenation of the contents of the first message (which while having no value in MSH-14, did end with DSC, and hence was actually a message fragment), plus all subsequent fragments (as identified by values in MSH-14).

f) If enhanced mode acknowledgments are used to request an accept ACK, then the receiver will acknowledge each fragment with an ACK message. Since each fragment has its own Message Control ID, each fragment level accept ACK will use the Message Control ID from the fragment it is acknowledging.

g) If enhanced mode acknowledgments are used to request an application level ACK, then the receiver will send an acknowledgment after receiving the final fragment.

Note:	The application level ACK should refer to the message by the Message Control ID of the first fragment.

Note:	The receiver can tell that a given incoming message is a fragment by the presence of the trailing DSC. Subsequent HL7 messages are identified as fragments by the presence of an MSH-14 value. The presence of a DSC in a fragment indicates that more fragments are to follow.

It is a protocol error to end a message with DSC, and then never send a fragment.

For example, a single logical message may be fragmented into three HL7 messages.

```
---- Sender HL7 message (incomplete,fragment1)---
MSH|||||||||1001||2.4|123||..
A|…
B|…
DSC|W4xy

---- Sender HL7 message (fragment 2)---
MSH|||||||||2106||2.4|124|W4xy|
C|…
D|…
DSC|V292
----- another HL7 message(fragment 3, final)---
MSH|||||||||2401||2.4|125|V292
E|…
```

Such a sequence is logically the same as the single message

```
MSH|….|2.4|123||..
A|…
B|…
C|…
D|…
E|…
```

See example in section 2.18.4 for a more elaborate example.

2.15.2.3 Segment fragmentation across messages

If the last segment of a fragment itself needs to be broken, then the following idiomatic use of ADD shall apply.

a) the segment being continued (call it ANY for this example) is ended at an arbitrary character position and terminated with the standard segment terminator (carriage return).

b) the following segment is the ADD segment. It will contain no characters other than “ADD”. (The lack of characters signals the receiver that ANY will be continued.)

c) The second following segment will be the DSC, used as described above in Section 2.15.2.2, "Segment fragmentation/continuation using the DSC segment ".

d) The first segment of the following fragment will be an ADD segment. The characters of this ADD segment are logically part of the ANY segment of the previous fragment.

For example

```
MSH|…|2.4|
A|12
ADD
DSC|JR97
--------- (fragment 2)
MSH|…|2.4|JR97
ADD|345
```

is logically the same as

```
MSH|…|2.4
A|12345
```

e) transaction flow for a continued unsolicited message with a continued segment. first unsolicited message and acknowledgment:

```
MSH
URD
[ URS ]
{DSP}                    (last DSP is incomplete)
ADD                      (contains no fields)
DSC                      (Continuation segment)

MSH                      (General acknowledgment)
MSA
[ ERR ]
```

f) second unsolicited message and acknowledgment:

```
MSH                      (contains continuation pointer from DSC segment of prior
                         message)
ADD                      (contains remainder of data from continued DSP segment
                         from prior message)
   {DSP}
```

Note:	This second message could itself be continued with a second DSC and (if needed) a second ADD segment prior to it.

```
MSH                      (General acknowledgment)
```

```
MSA
[ERR]
```

2.15.3 HL7 batch protocol

There are instances when it is convenient to transfer a batch of HL7 messages. Common examples would be a batch of financial posting detail transactions (DFT's) sent from an ancillary to a financial system. Such a batch could be sent online using a common file transfer protocol, or offline via tape or diskette.

2.15.3.1 HL7 batch file structure

The structure of an HL7 batch file is given by the following (using the HL7 abstract message syntax)

```
[FHS]                    (file header segment)
{ [BHS]                  (batch header segment)
 { [MSH                  (zero or more HL7 messages)
    ....
    ....
    ....
] }
[BTS]                    (batch trailer segment)
}
[FTS]                    (file trailer segment)
```

Notes:

The sequence numbering protocol has a natural application in batch transfers. See the discussion of batch acknowledgments that follows.

Although a batch will usually consist of a single type of message, there is nothing in the definition that restricts a batch to only one message type.

The HL7 file and batch header and trailer segments are defined in exactly the same manner as the HL7 message segments. Hence the HL7 message construction rules of Section 2.11, "Message construction rules," can be used to encode and decode HL7 batch files.

There are only two cases in which an HL7 batch file may contain zero HL7 messages:

a) a batch containing zero HL7 messages may be sent to meet a requirement for periodic submission of batches when there are no messages to send.

b) a batch containing zero negative acknowledgment messages may be sent to indicate that all the HL7 messages contained in the batch being acknowledged are implicitly acknowledged. See Section 2.15.3.3, "Acknowledging batches."

2.15.3.2 Related segments and data usage

The following segments defined in Section 2.16, "MESSAGE CONTROL SEGMENTS" relate to the HL7 Batch Protocol:

BHS Batch Header

BTS Batch Trailer

FHS File Header

FTS File Trailer

The BTS segment contains a field, *BTS-3-batch totals*, which may have one or more totals drawn from fields within the individual messages. The method for computing such totals will be determined on a site or application basis unless explicitly stated in a functional chapter.

2.15.3.3 Acknowledging batches

In general, the utility of sending batches of data is that the data is accepted all at once, with errors processed on an exception basis. However, it is a permissible application of HL7 to acknowledge all messages. Several options for acknowledgment are given and will be chosen on an application basis. In these cases, the sequence numbering protocol can be useful to the applications processing the batch.

The options are:

a) all messages are acknowledged in the response batch.

b) the receiving system prints some form of batch control report, which is then dealt with manually by personnel at the sending system. No acknowledgments are performed by the protocol software.

c) an automated acknowledgment batch is created containing acknowledgment messages only for those messages containing errors. In this mode an empty acknowledgment batch may be created (i.e., an HL7 batch file without any HL7 acknowledgment messages).

In each case where there is a response batch, its format is a batch of individual messages. Each individual message is in the format defined for an online response in the chapters. Consider, for example, a batch that might be constructed to respond to a batch of Detailed Financial Transactions (Chapter 6). The messages in the response batch would consist entirely of ACK messages, since ACK is the response shown in Chapter 6.

When batches are retransmitted after the correction of errors, *BHS-12-reference batch control ID* should contain the batch control ID of the original batch.

2.15.3.4 Batch message as a query response

The HL7 query also can be used to query for a batch in the following manner:

a) use the value BB or BL of *QRD-5-deferred response type* to specify a batch response. The query will be acknowledged with a general acknowledgment as in the Deferred Access example above (see chapter 5)

b) in addition, insert into the batch file the QRD and QRF segments as follows:

```
[FHS]                    (file header segment)
{ [BHS]                  (batch header segment)
  [QRD]                  (the QRD and QRF define the

  [QRF]                  query that this batch is a response to)
  { MSH                  (one or more HL7 messages)
    ....
    ....
    ....
   }
   [BTS]                 (batch trailer segment)
   }
[FTS]                    (file trailer segment)
```

c) the acknowledgment of a batch is described in this chapter (see Section 2.15.3.3, “Acknowledging batches”).

2.15.4 Modes for updating via repeating segments

When groups of repeating segments appear within a message it is not obvious from the basic HL7 abstract message syntax how best to apply the new group of repeating segments on the receiving system. HL7 suggests two methods: the “snapshot” mode and the “action code/unique identifier” mode.

Background:

The segments which repeat in HL7 messages Patient Administration (ADT)/Financial Information messages (AL1, DG1, PR1, GT1, IN1, IN2, IN3, NK1, NTE) present a problem if the requirement is to update only part of the information previously sent. Prior to Version 2.3 of the Standard, all such repeating segments had to be sent again in the update, because there was no way to indicate which ones changed and which ones did not. For example, if two DG1 segments were sent originally (containing a primary and secondary diagnosis), and then if a tertiary diagnoses needed to be sent, the sending system had to send all diagnoses which were currently valid, that is, three DG1 segments (containing primary, secondary and tertiary diagnosis). This way of doing things is referred to as the "snapshot" mode. In this mode, everything (all repeating segments) must be sent with every subsequent message in the series of messages.

In the Order Entry, Observation Reporting, and Master Files chapters, action codes (e.g., order control codes and result status codes) and unique identifiers (e.g., placer and filler numbers) are currently specified as part of the ORC, OBR, OBX and MFE segments. So, except for the NTE segments, this problem exists mainly for the Patient Administration and Financial Management chapter segments.

For systems implementing Version 2.3 or higher, if a particular repeating segment can be updated by either of these two modes, the parties concerned will determine by agreement on a site-specific basis whether an interface will use the "snapshot" mode or the "action code/unique identifier" mode.

2.15.4.1 Snapshot mode update definition

In the "snapshot" mode, the group of repeating segments from the incoming message replaces the prior group of repeating segments on the receiving system. This is equivalent to a deletion of the prior group followed by the addition of the new group. The snapshot mode is the usual mode in Version 2.2 and 2.1 implementations of HL7, but it is also available for Version 2.3 and future versions. To specify "delete all of the segments in this repeating group" in the snapshot mode, send a single segment with "delete data" indicated for all fields.

For example, if the following DG1 segment is in an ADT update message (for an inpatient stay):

```
DG1|""|""|""|""|""|""|""|""|""|""|""|""|""|""|""|""|""|""|""|""|""<cr>
```

and the snapshot mode is being used, this indicates that all previously-transmitted diagnoses for this inpatient stay should be deleted.

2.15.4.2 Action code/unique identifier mode update definition

In the "action code/unique identifier" mode, each member of a repeating group of segments must have a unique identifier (equivalent to the filler number in observational reports messages). The choice of delete/update/insert is determined by the action code (equivalent to the result status in observational reports messages). Refer to *HL7 Table 0206 - Segment action code* for valid values.

HL7 Table 0206 - Segment action code

Value	Description
A	Add/Insert
D	Delete
U	Update

The *unique identifier* is defined in a general manner as follows: it uniquely identifies one of multiple repetitions of the primary entity defined by the repeating segment in a way that does not change over time. It is not dependent on any particular message identifier level (MSH) fields; it functions across messages, not just within a message. The *unique identifier* will be chosen on a segment-specific basis, depending on the primary entity referenced by the segment. For some cases, such as a diagnosis code, it may be a CE data

type. For others, such as a person identifier, it may be a CX data type. For others it may be an EI (entity identifier) data type.

Note:	This mode is available for use only for new segments for Version 2.3 and for new segments in future versions

2.16 MESSAGE CONTROL SEGMENTS

The following segments are necessary to support the functionality described in this chapter.

Note:	The HL7 message construction rules define the standard HL7 encoding rules, creating variable length delimited messages from the segments defined below. Although only one set of encoding rules is defined as a standard in HL7 Version 2.3, other encoding rules are possible (but since they are non-standard, they may only used by a site-specific agreement).

The segments in this section are listed in alphabetical order. The following chart shows a summary of the segments listed by category.

Figure 2-8. HL7 message segments

Segment Category	Segment Name	HL7 Section Reference
Control		
	ADD	2.16.1
	BHS	2.16.2
	BTS	2.16.3
	DSC	2.16.4
	ERR	2.16.5
	FHS	2.16.6
	FTS	2.16.7
	MSA	2.16.8
	MSH	2.16.9
General Purpose		
	NTE	2.16.10
Query		
	All query segments have been moved to chapter 5	

2.16.1 ADD - addendum segment

The ADD segment is used to define the continuation of the prior segment in a continuation message. See Section2.15.2, “Continuation messages and segments,” for details.

HL7 Attribute Table - ADD – Addendum

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM#	ELEMENT NAME
1-n	65536	ST	O			00066	Addendum Continuation Pointer

2.16.1.0 ADD field definition

2.16.1.1 ADD-1 Addendum continuation pointer (ST) 00066

Definition: This field is used to define the continuation of the prior segment in a continuation message. See section 2.15.2, "Continuation messages and segments" for details. When the ADD is sent after the segment being continued, it contains no fields. It is only a marker that the previous segment is being continued in a subsequent message. Thus fields 1-N are not present. The sequence designation, 1-N, means the remainder of the fields in the segment being continued. These remainder of the segment being continued fields are present only when the ADD is sent with a continuation message.

2.16.2 BHS - batch header segment

The BHS segment defines the start of a batch.

HL7 Attribute Table - BHS – Batch Header

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	1	ST	R			00081	Batch Field Separator
2	3	ST	R			00082	Batch Encoding Characters
3	15	ST	O			00083	Batch Sending Application
4	20	ST	O			00084	Batch Sending Facility
5	15	ST	O			00085	Batch Receiving Application
6	20	ST	O			00086	Batch Receiving Facility
7	26	TS	O			00087	Batch Creation Date/Time
8	40	ST	O			00088	Batch Security
9	20	ST	O			00089	Batch Name/ID/Type
10	80	ST	O			00090	Batch Comment
11	20	ST	O			00091	Batch Control ID
12	20	ST	O			00092	Reference Batch Control ID

2.16.2.0 BHS field definitions

2.16.2.1 BHS-1 Batch field separator (ST) 00081

Definition: This field contains the separator between the segment ID and the first real field, *BHS-2-batch encoding characters*. As such it serves as the separator and defines the character to be used as a separator for the rest of the message. Recommended value is |,(ASCII 124).

2.16.2.2 BHS-2 Batch encoding characters (ST) 00082

Definition: This field contains the four characters in the following order: the component separator, repetition separator, escape characters, and subcomponent separator. Recommended values are ^~\& (ASCII 94, 126, 92, and 38, respectively). See Section 2.8, "MESSAGE DELIMITERS."

2.16.2.3 BHS-3 Batch sending application (ST) 00083

Definition: This field uniquely identifies the sending application among all other applications within the network enterprise. The network enterprise consists of all those applications that participate in the exchange of HL7 messages within the enterprise. Entirely site-defined.

2.16.2.4 BHS-4 Batch sending facility (ST) 00084

Definition: This field contains the address of one of several occurrences of the same application within the sending system. Absent other considerations, the Medicare Provider ID might be used with an appropriate sub-identifier in the second component. Entirely user-defined.

2.16.2.5 BHS-5 Batch receiving application (ST) 00085

Definition: This field uniquely identifies the receiving applications among all other applications within the network enterprise. The network enterprise consists of all those applications that participate in the exchange of HL7 messages within the enterprise. Entirely site-defined.

2.16.2.6 BHS-6 Batch receiving facility (ST) 00086

Definition: This field identifies the receiving application among multiple identical instances of the application running on behalf of different organizations. See comments *BHS-4-batch sending facility*. Entirely site-defined.

2.16.2.7 BHS-7 Batch creation date/time (TS) 00087

Definition: This field contains the date/time that the sending system created the message. If the time zone is specified, it will be used throughout the message as the default time zone.

2.16.2.8 BHS-8 Batch security (ST) 00088

Definition: In some applications of HL7, this field is used to implement security features. Its use is not yet further specified.

2.16.2.9 BHS-9 Batch name/ID/type (ST) 00089

Definition: This field can be used by the application processing the batch. It can have extra components if needed.

2.16.2.10 BHS-10 Batch comment (ST) 00090

Definition: This field is a comment field that is not further defined in the HL7 protocol.

2.16.2.11 BHS-11 Batch control ID (ST) 00091

Definition: This field is used to uniquely identify a particular batch. It can be echoed back in *BHS-12-reference batch control ID* if an answering batch is needed.

2.16.2.12 BHS-12 Reference batch control ID (ST) 00092

Definition: This field contains the value of *BHS-11-batch control ID* when this batch was originally transmitted. Not present if this batch is being sent for the first time. See definition for *BHS-11-batch control ID*.

2.16.3 BTS - batch trailer segment

The BTS segment defines the end of a batch.

HL7 Attribute Table - BTS – Batch Trailer

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	10	ST	O			00093	Batch Message Count
2	80	ST	O			00090	Batch Comment

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
3	100	NM	O	Y		00095	Batch Totals

2.16.3.0 BTS field definitions

2.16.3.1 BTS-1 Batch message count (ST) 00093

Definition: This field contains the count of the individual messages contained within the batch.

2.16.3.2 BTS-2 Batch comment (ST) 00090

Definition: This field is a comment field that is not further defined in the HL7 protocol.

2.16.3.3 BTS-3 Batch totals (NM) 00095

Definition: We encourage new users of this field to use the HL7 Version 2.3 data type of NM and to define it as "repeating." This field contains the batch total. If more than a single batch total exists, this field may be repeated.

This field may be defined as a CM data type for backward compatibility with HL7 Versions 2.2 and 2.1 with each total being carried as a separate component. Each component in this case is an NM data type.

2.16.4 DSC - continuation pointer segment

The DSC segment is used in the continuation protocol.

HL7 Attribute Table - DSC – Continuation Pointer

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	180	ST	O			00014	Continuation Pointer
2	1	ID	O		0398	01354	Continuation Style

2.16.4.0 DSC field definitions

2.16.4.1 DSC-1 Continuation pointer (ST) 00014

Definition: This field contains the continuation pointer. In an initial query, this field is not present. If the responder returns a value of null or not present, then there is no more data to fulfill any future continuation requests. For use with continuations of unsolicited messages, see chapter 5 and section 2.15.2, "Continuation messages and segments." Note that continuation protocols work with both display- and record-oriented messages.

2.16.4.2 DSC-2 Continuation style (ID) 01354

Definition: Indicates whether this is a fragmented message (see Section 2.15.2, "Continuation messages and segments"), or if it is part of an interactive continuation message (see Section 5.6.3, "Interactive continuation of response messages").

Refer to *HL7 Table 0398 – Continuation style* code for valid values.

HL7 Table 0398 - Continuation style code

Value	Description
F	Fragmentation
I	Interactive Continuation

2.16.5 ERR - error segment

The ERR segment is used to add error comments to acknowledgment messages.

HL7 Attribute Table - ERR –Error

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	80	CM	R	Y		00024	Error Code and Location

2.16.5.0 ERR field definition

2.16.5.1 ERR-1 Error code and location (CM) 00024

Components: <segment ID (ST)> ^ <sequence (NM)> ^ <field position (NM)> ^ <code identifying error (CE)>

Definition: This field identifies an erroneous segment in another message. The second component is an index if there is more than one segment of type <segment ID>. For systems that do not use the HL7 Encoding Rules, the data item number may be used for the third component. The fourth component (which references *HL7 Table 0357 - Message error condition codes*, (as a CE data type)) is restricted from having any subcomponents as the subcomponent separator is now the CE's component separator.

Note: See section 2.16.8.6, MSA-6-*error condition,* for a listing of this table.

2.16.6 FHS - file header segment

The FHS segment is used to head a file (group of batches) as defined in Section 2.15.3, "HL7 batch protocol."

HL7 Attribute Table - FHS - File Header

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	1	ST	R			00067	File Field Separator
2	4	ST	R			00068	File Encoding Characters
3	15	ST	O			00069	File Sending Application
4	20	ST	O			00070	File Sending Facility
5	15	ST	O			00071	File Receiving Application
6	20	ST	O			00072	File Receiving Facility
7	26	TS	O			00073	File Creation Date/Time
8	40	ST	O			00074	File Security
9	20	ST	O			00075	File Name/ID
10	80	ST	O			00076	File Header Comment
11	20	ST	O			00077	File Control ID
12	20	ST	O			00078	Reference File Control ID

2.16.6.0 FHS field definitions

2.16.6.1 FHS-1 File field separator (ST) 00067

Definition: This field has the same definition as the corresponding field in the MSH segment.

2.16.6.2 FHS-2 File encoding characters (ST) 00068

Definition: This field has the same definition as the corresponding field in the MSH segment.

2.16.6.3 FHS-3 File sending application (ST) 00069

Definition: This field has the same definition as the corresponding field in the MSH segment.

2.16.6.4 FHS-4 File sending facility (ST) 00070

Definition: This field has the same definition as the corresponding field in the MSH segment.

2.16.6.5 FHS-5 File receiving application (ST) 00071

Definition: This field has the same definition as the corresponding field in the MSH segment.

2.16.6.6 FHS-6 File receiving facility (ST) 00072

Definition: This field has the same definition as the corresponding field in the MSH segment.

2.16.6.7 FHS-7 File creation date/time (TS) 00073

Definition: This field has the same definition as the corresponding field in the MSH segment.

2.16.6.8 FHS-8 File security (ST) 00074

Definition: This field has the same definition as the corresponding field in the MSH segment.

2.16.6.9 FHS-9 File name/ID (ST) 00075

Definition: This field can be used by the application processing file. Its use is not further specified.

2.16.6.10 FHS-10 File header comment (ST) 00076

Definition: This field contains the free text field, the use of which is not further specified.

2.16.6.11 FHS-11 File control ID (ST) 00077

Definition: This field is used to identify a particular file uniquely. It can be echoed back in *FHS-12-reference file control ID.*

2.16.6.12 FHS-12 Reference file control ID (ST) 00078

Definition: This field contains the value of *FHS-11-file control ID* when this file was originally transmitted. Not present if this file is being transmitted for the first time.

2.16.7 FTS - file trailer segment

The FTS segment defines the end of a file.

HL7 Attribute Table - FTS - File Trailer

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	10	NM	O			00079	File Batch Count
2	80	ST	O			00080	File Trailer Comment

2.16.7.0 FTS field definitions

2.16.7.1 FTS-1 File batch count (NM) 00079

Definition: This field contains the number of batches contained in this file.

2.16.7.2 FTS-2 File trailer comment (ST) 00080

Definition: The use of this free text field is not further specified.

2.16.8 MSA - message acknowledgment segment

The MSA segment contains information sent while acknowledging another message.

HL7 Attribute Table - MSA - Message Acknowledgment

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	2	ID	R		0008	00018	Acknowledgment Code
2	20	ST	R			00010	Message Control ID
3	80	ST	O			00020	Text Message
4	15	NM	O			00021	Expected Sequence Number
5	1	ID	B		0102	00022	Delayed Acknowledgment Type
6	250	CE	O		0357	00023	Error Condition

2.16.8.0 MSA field definitions

2.16.8.1 MSA-1 Acknowledgment code (ID) 00018

Definition: This field contains an acknowledgment code, see message processing rules. Refer to *HL7 Table 0008 - Acknowledgment code* for valid values.

HL7 Table 0008 - Acknowledgment code

Value	Description
AA	Original mode: Application Accept - Enhanced mode: Application acknowledgment: Accept
AE	Original mode: Application Error - Enhanced mode: Application acknowledgment: Error
AR	Original mode: Application Reject - Enhanced mode: Application acknowledgment: Reject
CA	Enhanced mode: Accept acknowledgment: Commit Accept
CE	Enhanced mode: Accept acknowledgment: Commit Error
CR	Enhanced mode: Accept acknowledgment: Commit Reject

2.16.8.2 MSA-2 Message control ID (ST) 00010

Definition: This field contains the message control ID of the message sent by the sending system. It allows the sending system to associate this response with the message for which it is intended.

2.16.8.3 MSA-3 Text message (ST) 00020

Definition: This optional field further describes an error condition. This text may be printed in error logs or presented to an end user.

Use of *MSA-3-text message* and *MSA-6-error condition* are deprecated in favor of *ERR-1-Error code and location*. The ERR segment allows for richer descriptions of the erroneous conditions.

2.16.8.4 MSA-4 Expected sequence number (NM) 00021

Definition: This optional numeric field is used in the sequence number protocol.

2.16.8.5 MSA-5 Delayed acknowledgment type (ID) 00022

Definition: ***This field has been retained for backward compatibility.*** This field is used only as described above, in Section 2.13.2, "Application (level 7) processing rules, deferred processing two phase reply (original acknowledgment mode only)." Otherwise this field is not used.

HL7 Table 0102 - Delayed acknowledgment type

Value	Description
D	Message received, stored for later processing
F	acknowledgment after processing

2.16.8.6 MSA-6 Error condition (CE) 00023

```
Components: <identifier (ST)> ^ <text (ST)> ^ <name of coding system (IS)> ^ <alternate identifier
            (ST)> ^ <alternate text (ST)> ^ <name of alternate coding system (IS)>
```

Definition: This field allows the acknowledging system to use a user-defined error code to further specify AR or AE type acknowledgments. This field is a generalized replacement for *MSA-3-text message*.

Use of *MSA-3-text message* and *MSA-6-error condition* are deprecated in favor of *ERR-1-Error code and location*. The ERR segment allows for richer descriptions of the erroneous conditions.

The Message Error Condition codes are defined by *HL7 Table 0357 - Message error condition codes*.

HL7 Table 0357 - Message error condition codes

Error Condition Code	Error Condition Text	Description/Comment
Success		
0	Message accepted	Success. Optional, as the AA conveys success. Used for systems that must always return a status code.
Errors		
100	Segment sequence error	The message segments were not in the proper order, or required segments are missing.
101	Required field missing	A required field is missing from a segment
102	Data type error	The field contained data of the wrong data type, e.g. an NM field contained "FOO".
103	Table value not found	A field of data type ID or IS was compared against the corresponding table, and no match was found.
Rejection		
200	Unsupported message type	The Message Type is not supported.
201	Unsupported event code	The Event Code is not supported.
202	Unsupported processing id	The Processing ID is not supported.
203	Unsupported version id	The Version ID is not supported.
204	Unknown key identifier	The ID of the patient, order, etc., was not found. Used for transactions *other than* additions, e.g. transfer of a non-existent patient.
205	Duplicate key identifier	The ID of the patient, order, etc., already exists. Used in response to addition transactions (Admit, New Order, etc.).
206	Application record locked	The transaction could not be performed at the application storage level, e.g.

Error Condition Code	Error Condition Text	Description/Comment
		database locked.
207	Application internal error	A catchall for internal errors not explicitly covered by other codes.

2.16.9 MSH - message header segment

The MSH segment defines the intent, source, destination, and some specifics of the syntax of a message.

HL7 Attribute Table - MSH - Message Header

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	1	ST	R			00001	Field Separator
2	4	ST	R			00002	Encoding Characters
3	180	HD	O		0361	00003	Sending Application
4	180	HD	O		0362	00004	Sending Facility
5	180	HD	O		0361	00005	Receiving Application
6	180	HD	O		0362	00006	Receiving Facility
7	26	TS	R			00007	Date/Time Of Message
8	40	ST	O			00008	Security
9	13	CM	R		0076/ 0003	00009	Message Type
10	20	ST	R			00010	Message Control ID
11	3	PT	R			00011	Processing ID
12	60	VID	R		0104	00012	Version ID
13	15	NM	O			00013	Sequence Number
14	180	ST	O			00014	Continuation Pointer
15	2	ID	O		0155	00015	Accept Acknowledgment Type
16	2	ID	O		0155	00016	Application Acknowledgment Type
17	3	ID	O		0399	00017	Country Code
18	16	ID	O	Y	0211	00692	Character Set
19	250	CE	O			00693	Principal Language Of Message
20	20	ID	O		0356	01317	Alternate Character Set Handling Scheme
21	10	ID	O	Y	0449	01598	Conformance Statement ID

2.16.9.0 MSH field definitions

2.16.9.1 MSH-1 Field separator (ST) 00001

Definition: This field contains the separator between the segment ID and the first real field, *MSH-2-encoding characters*. As such it serves as the separator and defines the character to be used as a separator for the rest of the message. Recommended value is |, (ASCII 124).

2.16.9.2 MSH-2 Encoding characters (ST) 00002

Definition: This field contains the four characters in the following order: the component separator, repetition separator, escape character, and subcomponent separator. Recommended values are ^~\& (ASCII 94, 126, 92, and 38, respectively). See Section 2.8, "MESSAGE DELIMITERS."

2.16.9.3 MSH-3 Sending application (HD) 00003

```
Components:  <namespace ID (IS)> ^ <universal ID (ST)> ^ <universal ID type (ID)>
```

Definition: This field uniquely identifies the sending application among all other applications within the network enterprise. The network enterprise consists of all those applications that participate in the exchange of HL7 messages within the enterprise. Entirely site-defined. *User-defined Table 0361-Sending/receiving application* is used as the user-defined table of values for the first component.

User-defined Table 0361 – Sending/receiving application

Value	Description
	No suggested values defined

Note: By site agreement, implementors may continue to use *User-defined Table 0300 – Namespace ID* for the first component.

2.16.9.4 MSH-4 Sending facility (HD) 00004

```
Components:  <namespace ID (IS)> ^ <universal ID (ST)> ^ <universal ID type (ID)>
```

Definition: This field further describes the sending application, *MSH-3-sending application*. With the promotion of this field to an HD data type, the usage has been broadened to include not just the sending facility but other organizational entities such as a) the organizational entity responsible for sending application; b) the responsible unit; c) a product or vendor's identifier, etc. Entirely site-defined. *User-defined Table 0362 – Sending/receiving facility* is used as the HL7 identifier for the user-defined table of values for the first component.

User-defined Table 0362 – Sending/receiving facility

Value	Description
	No suggested values defined

Note: By site agreement, implementers may continue to use *User-defined Table 0300 – Namespace ID* for the first component.

2.16.9.5 MSH-5 Receiving application (HD) 00005

```
Components:  <namespace ID (IS)> ^ <universal ID (ST)> ^ <universal ID type (ID)>
```

Definition: This field uniquely identifies the receiving application among all other applications within the network enterprise. The network enterprise consists of all those applications that participate in the exchange of HL7 messages within the enterprise. Entirely site-defined. *User-defined Table 0361-Sending/receiving application* is used as the HL7 identifier for the user-defined table of values for the first component.

Note: By site agreement, implementers may continue to use *User-defined Table 0300 – Namespace ID* for the first component.

2.16.9.6 MSH-6 Receiving facility (HD) 00006

```
Components:  <namespace ID (IS)> ^ <universal ID (ST)> ^ <universal ID type (ID)>
```

Definition: This field identifies the receiving application among multiple identical instances of the application running on behalf of different organizations. *User-defined Table 0362 – Sending/receiving facility* is used as the HL7 identifier for the user-defined table of values for the first component. Entirely site-defined.

Note: By site agreement, implementers may continue to use *User-defined Table 0300 – Namespace ID* for the first component.

2.16.9.7 MSH-7 Date/time of message (TS) 00007

Definition: This field contains the date/time that the sending system created the message. If the time zone is specified, it will be used throughout the message as the default time zone.

Note: This field was made required in version 2.4. Messages with versions prior to 2.4 are not required to value this field. This usage supports backward compatibility.

2.16.9.8 MSH-8 Security (ST) 00008

Definition: In some applications of HL7, this field is used to implement security features. Its use is not yet further specified.

2.16.9.9 MSH-9 Message type (CM) 00009

```
Components:  <message type (ID)> ^ <trigger event (ID)> ^ <message structure (ID)>
```

Definition: This field contains the message type, trigger event, and the message structure ID for the message.

The first component is the message type code defined by *HL7 Table 0076 - Message type*. This table contains values such as ACK, ADT, ORM, ORU etc. See section 2.17.1 or Appendix A for complete listing.

The second component is the trigger event code defined by *HL7 Table 0003 - Event type*. This table contains values like A01, O01, R01 etc. See section 2.17.2 or Appendix A for a complete listing

The third component is the abstract message structure code defined by *HL7 Table 0354 - Message structure*. This table has two columns. The first column contains the value of this code, which describes a particular HL7 “abstract message structure definition” in terms of segments, as defined in sections 2.12, “CHAPTER FORMATS FOR DEFINING HL7 MESSAGES” and 2.12.1, “HL7 abstract message syntax example”. The second column of table 0354 lists the various HL7 trigger events that use the particular abstract message definition. For example, the message structure code ADT_A01 describes the single abstract message structure used by the trigger events A01, A04, A05, A08, A13, A14, A28 and A31. See section 2.17.3 or Appendix A for a complete listing.

The receiving system uses this field to recognize the data segments, and possibly, the application to which to route this message. For certain queries, which may have more than a single response event type, the second component may, in the response message, vary to indicate the response event type. See the discussion of the display query variants in chapter 5. The second component is not required on response or acknowledgment messages.

2.16.9.10 MSH-10 Message control ID (ST) 00010

Definition: This field contains a number or other identifier that uniquely identifies the message. The receiving system echoes this ID back to the sending system in the Message acknowledgment segment (MSA).

2.16.9.11 MSH-11 Processing ID (PT) 00011

```
Components:  <processing ID (ID)> ^ <processing mode (ID)>
```

Definition: This field is used to decide whether to process the message as defined in HL7 Application (level 7) Processing rules. The first component defines whether the message is part of a production, training, or debugging system (refer to *HL7 Table 0103 - Processing ID* for valid values). The second component defines whether the message is part of an archival process or an initial load (refer to *HL7 Table 0207 - Processing mode* for valid values). This allows different priorities to be given to different processing modes.

HL7 Table 0103 - Processing ID

Value	Description
D	Debugging
P	Production
T	Training

HL7 Table 0207 - Processing mode

Value	Description
A	Archive
R	Restore from archive
I	Initial load
T	Current processing, transmitted at intervals (scheduled or on demand)
Not present	Not present (the default, meaning *current* processing)

2.16.9.12 MSH-12 Version ID (VID) 00012

```
Components:  <version ID (ID)> ^ <internationalization code (CE)> ^ <internal version ID (CE)>
```

Definition: This field is matched by the receiving system to its own version to be sure the message will be interpreted correctly. Beginning with Version 2.3.1, it has two additional "internationalization" components, for use by HL7 international affiliates. The <internationalization code> is CE data type (using the ISO country codes where appropriate) which represents the HL7 affiliate. The <internal version ID> is used if the HL7 Affiliate has more than a single 'local' version associated with a single US version. The <internal version ID> has a CE data type, since the table values vary for each HL7 Affiliate.

HL7 Table 0104 - Version ID

Value	Description	Date
2.0	Release 2.0	September 1988
2.0D	Demo 2.0	October 1988
2.1	Release 2. 1	March 1990
2.2	Release 2.2	December 1994
2.3	Release 2.3	March 1997
2.3.1	Release 2.3.1	May 1999
2.4	Release 2.4	November 2000

2.16.9.13 MSH-13 Sequence number (NM) 00013

Definition: A non-null value in this field implies that the sequence number protocol is in use. This numeric field is incremented by one for each subsequent value.

2.16.9.14 MSH-14 Continuation pointer (ST) 00014

Definition: This field is used to define continuations in application-specific ways.

Only the sender of a fragmented message values this field.

2.16.9.15 MSH-15 Accept acknowledgment type (ID) 00015

Definition: This field identifies the conditions under which accept acknowledgments are required to be returned in response to this message. Required for enhanced acknowledgment mode. Refer *to HL7 Table 0155 - Accept/application acknowledgment conditions* for valid values.

2.16.9.16 MSH-16 Application acknowledgment type (ID) 00016

Definition: This field contains the conditions under which application acknowledgments are required to be returned in response to this message. Required for enhanced acknowledgment mode.

The following table contains the possible values for *MSH-15-accept acknowledgment type* and *MSH-16-application acknowledgment type*:

HL7 Table 0155 - Accept/application acknowledgment conditions

Value	Description
AL	Always
NE	Never
ER	Error/reject conditions only
SU	Successful completion only

Note: If *MSH-15-accept acknowledgment type* and *MSH-16-application acknowledgment type* are omitted (or are both null), the original acknowledgment mode rules are used.

2.16.9.17 MSH-17 Country code (ID) 00017

Definition: This field contains the country of origin for the message. It will be used primarily to specify default elements, such as currency denominations. The values to be used are those of ISO 3166, which are reprinted here upon written approval from ANSI.[2]. The ISO 3166 table has three separate forms of the country code: HL7 specifies that the 3-character (alphabetic) form be used for the country code.

Refer to *HL7 Table 0399 – Country code* for the 3-character codes as defined by ISO 3166 table.

HL7 Table 0399 – Country code

Value	Description
ABW	ARUBA
AFG	AFGHANISTAN
AFT	FRENCH SOUTHERN TERRITORIES
AGO	ANGOLA
AIA	ANGUILLA
ALB	ALBANIA
AND	ANDORRA
ANT	NETHERLANDS ANTILLES
ARE	UNITED ARAB EMIRATES
ARG	ARGENTINA
ARM	ARMENIA
ASM	AMERICAN SAMOA

[2] Available from ISO 1 Rue de Varembe, Case Postale 56, CH 1211, Geneve, Switzerland

Value	Description
ATA	ANTARCTICA
ATG	ANTIGUA AND BARBUDA
AUS	AUSTRALIA
AUT	AUSTRIA
AZE	AZERBAIJAN
BDI	BURUNDI
BEL	BELGIUM
BEN	BENIN
BFA	BURKINA FASO
BGD	BANGLADESH
BGR	BULGARIA
BHR	BAHRAIN
BHS	BAHAMAS
BIH	BOSNIA AND HERZEGOVINA
BLR	BELARUS
BLZ	BELIZE
BMU	BERMUDA
BOL	BOLIVIA
BRA	BRAZIL
BRB	BARBADOS
BRN	BRUNEI DARUSSALAM
BTN	BHUTAN
BVT	BOUVET ISLAND
BWA	BOTSWANA
CAF	CENTRAL AFRICAN REPUBLIC
CAN	CANADA
CCK	COCOS (KEELING) ISLANDS
CHE	SWITZERLAND
CHL	CHILE
CHN	CHINA
CIV	COTE D'VOIRE
CMR	CAMEROON
COD	CONGO, THE DEMOCRATIC REPUBLIC OF THE
COG	CONGO
COK	COOK ISLAND
COL	COLOMBIA
COM	COMOROS
CPV	CAPE VERDE
CRI	COSTA RICA
CUB	CUBA
CXR	CHRISTMAS ISLAND
CYM	CAYMAN ISLANDS
CYP	CYPRUS
CZE	CZECH REPUBLIC
DEU	GERMANY
DJI	DJIBOUTI

Value	Description
DMA	DOMINICA
DNK	DENMARK
DOM	DOMINICAN REPUBLIC
DZA	ALGERIA
ECU	ECUADOR
EGY	EGYPT
ERI	ERITREA
ESH	WESTERN SAHARA
ESP	SPAIN
EST	ESTONIA
ETH	ETHIOPIA
FIN	FINLAND
FJI	FIJI
FLK	FALKLAND ISLANDS (MALVINAS)
FRA	FRANCE
FRO	FAROE ISLANDS
FSM	MICRONESIA, FEDERATED STATES OF
GAB	GABON
GBR	UNITED KINGDOM
GEO	GEORGIA
GHA	GHANA
GIB	GIBRALTAR
GIN	GUINEA
GLP	GUADELOUPE
GMB	GAMBIA
GNB	GUINEA-BISSAU
GNQ	EQUATORIAL GUINEA
GRC	GREECE
GRD	GRENADA
GRL	GREENLAND
GTM	GUATEMALA
GUF	FRENCH GUIANA
GUM	GUAM
GUY	GUYANA
HKG	HONG KONG
HMD	HEARD ISLAND AND MCDONALD ISLANDS
HND	HONDURAS
HRV	CROATIA
HTI	HAITI
HUN	HUNGARY
IDN	INDONESIA
IND	INDIA
IOT	BRITISH INDIAN OCEAN TERRITORY
IRL	IRELAND
IRN	IRAN, ISLAMIC REPUBLIC OF
IRQ	IRAQ

Value	Description
ISL	ICELAND
ISR	ISRAEL
ITA	ITALY
JAM	JAMAICA
JOR	JORDAN
JPN	JAPAN
KAZ	KAZAKSTAN
KEN	KENYA
KGZ	KYRGYZSTAN
KHM	CAMBODIA
KIR	KIRIBATI
KNA	SAINT KITTS AND NEVIS
KOR	KOREA, REPUBLIC OF
KWT	KUWAIT
LAO	LAO PEOPLE'S DEMOCRATIC REPUBLIC
LBN	LEBANNON
LBR	LIBERIA
LBY	LIBYAN ARAB JAMAHIRIYA
LCA	SAINT LUCIA
LIE	LIECHTENSTEIN
LKA	SRI LANKA
LSO	LESOTHO
LTU	LITHUANIA
LUX	LUXEMBOURG
LVA	LATIVA
MAC	MACAU
MAR	MOROCCO
MCO	MONACO
MDA	MOLDOVA, REPUBLIC OF
MDG	MADAGASCAR
MDV	MALDIVES
MEX	MEXICO
MHL	MARSHALL ISLANDS
MKD	MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF
MLI	MALI
MLT	MALTA
MMR	MYANMAR
MNG	MONGOLIA
MNP	NORTHERN MARIANA ISLANDS
MOZ	MOZAMBIQUE
MRT	MAURITANIA
MSR	MONTSERRAT
MTQ	MARTINIQUE
MUS	MAURITUS
MWI	MALAWI
MYS	MALAYSIA

Value	Description
MYT	MAYOTTE
NAM	NAMIBIA
NCL	NEW CALEDONIA
NER	NIGER
NFK	NORFOLK ISLAND
NGA	NIGERIA
NIC	NICARAGUA
NIU	NIUE
NLD	NETHERLANDS
NOR	NORWAY
NPL	NEPAL
NRU	NAURU
NZL	NEW ZEALAND
OMN	OMAN
PAK	PAKISTAN
PAN	PANAMA
PCN	PITCAIRN
PER	PERU
PHL	PHILIPPINES
PLW	PALAU
PNG	PAPUA NEW GUINEA
POL	POLAND
PRI	PUERTO RICO
PRK	KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF
PRT	PORTUGAL
PRY	PARAGUAY
PYF	FRENCH POLYNESIA
QAT	QATAR
REU	REUNION
ROM	ROMANIA
RUS	RUSSIAN FEDERATION
RWA	RWANDA
SAU	SAUDI ARABIA
SDN	SUDAN
SEN	SENEGAL
SGP	SINGAPORE
SGS	SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS
SHN	SAINT HELENA
SJM	SVALBARD AND JAN MAYEN
SLB	SOLOMON ISLANDS
SLE	SIERRA LEONE
SLV	EL SALVADOR
SMR	SAN MARINO
SOM	SOMALIA
SPM	SAINT PIERRE AND MIQUELON
STP	SAO TOME AND PRINCIPE

Value	Description
SUR	SURINAME
SVK	SLOVAKIA
SVN	SLOVENIA
SWE	SWEDEN
SWZ	SWAZILAND
SYC	SEYCHELLES
SYR	SYRIAN ARAB REPUBLIC
TCA	TURKS AND CAICOS ISLANDS
TCD	CHAD
TGO	TOGO
THA	THAILAND
TJK	TAJIKISTAN
TKL	TOKELAU
TKM	TURKMENISTAN
TMP	EAST TIMOR
TON	TONGA
TTO	TRINIDAD AND TOBAGO
TUN	TUNISIA
TUR	TURKEY
TUV	TUVALU
TWN	TAIWAN, PROVINCE OF CHINA
TZA	TANZANIA, UNITED REPUBLIC OF
UGA	UGANDA
UKR	UKRAINE
UMI	UNITED STATES MINOR OUTLYING ISLANDS
URY	URUGUAY
USA	UNITED STATES
UZB	UZBEKISTAN
VAT	HOLY SEE (VATICAN CITY STATE)
VCT	SAINT VINCENT AND THE GRENADINES
VEN	VENEZUELA
VGB	VIRGIN ISLANDS, BRITISH
VIR	VIRGIN ISLANDS, U.S.
VNM	VIET NAM
VUT	VANUATU
WLF	WALLIS AND FUTUNA
WSM	SAMOA
YEM	YEMEN
YUG	YUGOSLAVIA
ZAF	SOUTH AFRICA
ZMB	ZAMBIA
ZWE	ZIMBABWE

2.16.9.18 MSH-18 Character set (ID) 00692

Definition: This field contains the character set for the entire message. Refer to *HL7 Table 0211 - Alternate character sets* for valid values.

HL7 Table 0211 - Alternate character sets

Value	Description
ASCII	The printable 7-bit ASCII character set. (This is the default if this field is omitted)
8859/1	The printable characters from the ISO 8859/1 Character set
8859/2	The printable characters from the ISO 8859/2 Character set
8859/3	The printable characters from the ISO 8859/3 Character set
8859/4	The printable characters from the ISO 8859/4 Character set
8859/5	The printable characters from the ISO 8859/5 Character set
8859/6	The printable characters from the ISO 8859/6 Character set
8859/7	The printable characters from the ISO 8859/7 Character set
8859/8	The printable characters from the ISO 8859/8 Character set
8859/9	The printable characters from the ISO 8859/9 Character set
ISO IR14	Code for Information Exchange (one byte)(JIS X 0201-1976). Note that the code contains a space, i.e. "ISO IR14".
ISO IR87	Code for the Japanese Graphic Character set for information interchange (JIS X 0208-1990), Note that the code contains a space, i.e. "ISO IR87".
ISO IR159	Code of the supplementary Japanese Graphic Character set for information interchange (JIS X 0212-1990). Note that the code contains a space, i.e. "ISO IR159".
UNICODE	The world wide character standard from ISO/IEC 10646-1-19933

Note: The field separator character must still be chosen from the printable 7-bit ASCII character set.

The repetitions of this field to specify different character sets apply only to fields of the, FT, ST, and TX data types.

The field *MSH-18-character set* is an optional, repeating field of data type ID, using IDs outlined in *HL7 Table 0211 - Alternate character sets* (or equivalents from "ISO 2375").

- if the field is not valued, the default single-byte character set (ASCII ("ISO IR6")) should be assumed. No other character sets are allowed in the message.
- if the field repeats, but the first element is NULL (i.e., present but unvalued), the single-byte ASCII ("ISO IR6") is assumed as the default character set.
- if the sequence is present and the first element is specified, this character set is regarded as the default character set for the message. This must be a single-byte character set (i.e., "ISO IR6", "ISO IR13", "ISO IR14", "ISO IR100", etc.).
- elements in the remainder of the sequence (i.e., elements 2..n) are alternate character sets that may be used. These may include multi-byte character sets (i.e., JIS X 0208).
- the default character set should always be a single-byte character set. It should always have "ISO IR6" (ISO 646) or "ISO IR14" (JIS X 0201-1976) in the G0 area.

[3] Available from The Unicode Consortium, P.O. Box 700519, San Jose, CA 95170-0519. See http://www/unicode.org/unicode/consortium/consort.html

2.16.9.19 MSH-19 Principal language of message (CE) 00693

```
Components: <identifier (ST)> ^ <text (ST)> ^ <name of coding system (IS)> ^ <alternate identifier
            (ST)> ^ <alternate text (ST)> ^ <name of alternate coding system (IS)>
```

Definition: This field contains the principal language of the message. Codes come from ISO 639.

2.16.9.20 MSH-20 Alternate character set handling scheme (ID) 01317

Definition: When any alternative character sets are used (as specified in the second or later components of *MSH-18 character sets*), and if any special handling scheme is needed, this component is to specify the scheme used, according to *HL7 Table 0356- Alternate character set handling scheme* as defined below:

HL7 Table 0356 - Alternate character set handling scheme

Value	Description
ISO 2022-1994	This standard is titled "Information Technology - Character Code Structure and Extension Technique". This standard specifies an escape sequence from basic one byte character set to specified other character set, and vice versa. The escape sequence explicitly specifies what alternate character set to be evoked. Note that in this mode, the actual ASCII escape character is used as defined in the referenced ISO document. As noted in 1.6.1., escape sequences to/from alternate character set should occur within HL7 delimiters. In other words, HL7 delimiters are basic one byte characters only, and just before and just after delimiters, character encoding status should be the basic one byte set.
2.3	The character set switching mode specified in HL7 2.3, sections 2.8.28.6.1, and 2.9.2. Note that the escape sequences used in this mode do not use the ASCII "esc" character. They are "HL7 escape sequences" as defined in HL7 2.3, sec. 2.9 as defined in ISO 2022-1994 (Also, note that sections 2.8.28.6.1and 2.9.2 in HL7 2.3 correspond to sections 2.8.31.6.1and 2.9.2 in HL7 2.4.)
<null>	This is the default, indicating that there is no character set switching occurring in this message.

2.16.9.21 MSH-21 Conformance statement ID (ID) 01598

Definition: Sites may use this field to assert adherence to a Conformance Statement published by HL7 or by a site. Conformance Statements contain detailed explanations of grammar, syntax, and usage for a particular message or set of messages. Examples of the use of Conformance Statements appear in Chapter 5, "Query."

Repetition of this field allows more flexibility in creating and naming conformance statements. For example, the first repetition could reference a standard conformance statement, and the second, just some changes to it.

Values for HL7-standard conformance statements appear in *HL7 Table 0449 - Conformance statements* as defined below.

HL7 Table 0449 - Conformance statements

Value	Description

Note: As HL7 technical committees ballot conformance statements, table 449 will be populated with their identifiers. No identifiers have been issued as of v 2.4. As with any HL7 table, this table may be extended with site-defined identifiers.

2.16.10 NTE - notes and comments segment

The NTE segment is defined here for inclusion in messages defined in other chapters. It is commonly used for sending notes and comments.

HL7 Attribute Table - NTE - Notes and Comments

SEQ	LEN	DT	OPT	RP/#	TBL#	ITEM #	ELEMENT NAME
1	4	SI	O			00096	Set ID - NTE
2	8	ID	O		0105	00097	Source of Comment
3	65536	FT	O	Y		00098	Comment
4	250	CE	O		0364	01318	Comment Type

2.16.10.0 NTE field definitions

2.16.10.1 NTE-1 Set ID - NTE (SI) 00096

Definition: This field may be used where multiple NTE segments are included in a message. Their numbering must be described in the application message definition.

2.16.10.2 NTE-2 Source of comment (ID) 00097

Definition: This field is used when source of comment must be identified. This table may be extended locally during implementation. Refer to *HL7 Table 0105 - Source of comment* for valid values.

HL7 Table 0105 - Source of comment

Value	Description
L	Ancillary (filler) department is source of comment
P	Orderer (placer) is source of comment
O	Other system is source of comment

2.16.10.3 NTE-3 Comment (FT) 00098

Definition: This field contains the comment contained in the segment.

Note: In the current HL7 version, this is a FT rather than a TX data type. Since there is no difference between a FT data type without any embedded formatting commands, and a TX data type, this change is compatible with the previous version.

2.16.10.4 NTE-4 Comment type (CE) 01318

```
Components: <identifier (ST)> ^ <text (ST)> ^ <name of coding system (IS)> ^ <alternate identifier
            (ST)> ^ <alternate text (ST)> ^ <name of alternate coding system (IS)>
```

Definition: This field contains a value to identify the type of comment text being sent in the specific comment record. Refer to *User-defined Table 0364 - Comment type* for suggested values.

User-defined Table 0364 - Comment type

Value	Description
PI	Patient Instructions
AI	Ancillary Instructions
GI	General Instructions
1R	Primary Reason
2R	Secondary Reason
GR	General Reason
RE	Remark

Value	Description
DR	Duplicate/Interaction Reason

Note: A field already exists on the NTE record that identifies the Sources of Comment (e.g., ancillary, placer, other). However some applications need to support other types of comment text (e.g., instructions, reason, remarks, etc.). A separate NTE segment can be used for each type of comment (e.g., instructions are on one NTE and remarks on another NTE).

2.17 MISCELLANEOUS HL7 TABLES USED ACROSS ALL CHAPTERS

2.17.1 Message type table

HL7 Table 0076 - Message type

Message	Description	Chapter
ACK	General acknowledgment message	2
ADR	ADT response	3
ADT	ADT message	3
BAR	Add/change billing account	6
CRM	Clinical study registration message	7
CSU	Unsolicited study data message	7
DFT	Detail financial transactions	6
DOC	Document response	9
DSR	Display response	5
EAC	Automated equipment command message	13
EAN	Automated equipment notification message	13
EAR	Automated equipment response message	13
EDR	Enhanced display response	2
EQQ	Embedded query language query	2
ERP	Event replay response	2
ESR	Automated equipment status update acknowledgment message	13
ESU	Automated equipment status update message	13
INR	Automated equipment inventory request message	13
INU	Automated equipment inventory update message	13
LSR	Automated equipment log/service request message	13
LSU	Automated equipment log/service update message	13
MCF	Delayed Acknowledgment (Retained for backward compatibility only)	2
MDM	Medical document management	9
MFD	Master files delayed application acknowledgment	8
MFK	Master files application acknowledgment	8
MFN	Master files notification	8
MFQ	Master files query	8

Message	Description	Chapter
MFR	Master files response	8
NMD	Application management data message	14
NMQ	Application management query message	14
NMR	Application management response message	14
OMD	Dietary order	4
OMG	General clinical order message	4
OML	Laboratory order message	4
OMN	Non-stock requisition order message	4
OMP	Pharmacy/treatment order message	4
OMS	Stock requisition order message	4
OMS	Stock requisition order message	4
ORD	Dietary order acknowledgment message	4
ORF	Query for results of observation	7
ORG	General clinical order acknowledgment message	4
ORL	Laboratory acknowledgment message (unsolicited)	7
ORM	Pharmacy/treatment order message	4
ORN	Non-stock requisition - General order acknowledgment message	4
ORP	Pharmacy/treatment order acknowledgment message	4
ORR	General order response message response to any ORM	4
ORS	Stock requisition - Order acknowledgment message	4
ORU	Unsolicited transmission of an observation message	7
OSQ	Query response for order status	4
OSR	Query response for order status	4
OUL	Unsolicited laboratory observation message	7
PEX	Product experience message	7
PGL	Patient goal message	12
PIN	Patient insurance information	11
PMU	Add personnel record	15
PPG	Patient pathway message (goal-oriented)	12
PPP	Patient pathway message (problem-oriented)	12
PPR	Patient problem message	12
PPT	Patient pathway goal-oriented response	12
PPV	Patient goal response	12
PRR	Patient problem response	12
PTR	Patient pathway problem-oriented response	12
QBP	Query by parameter	5
QCK	Deferred query	2

Message	Description	Chapter
QCN	Cancel query	5
QRY	Query, original mode	3
QSB	Create subscription	5
QSX	Cancel subscription/acknowledge message	5
QVR	Query for previous events	5
RAR	Pharmacy/treatment administration information	4
RAS	Pharmacy/treatment administration message	4
RCI	Return clinical information	11
RCL	Return clinical list	11
RDE	Pharmacy/treatment encoded order message	4
RDR	Pharmacy/treatment dispense information	4
RDS	Pharmacy/treatment dispense message	4
RDY	Display based response	5
REF	Patient referral	11
RER	Pharmacy/treatment encoded order information	4
RGR	Pharmacy/treatment dose information	4
RGV	Pharmacy/treatment give message	4
ROR	Pharmacy/treatment order response	4
RPA	Return patient authorization	11
RPI	Return patient information	11
RPL	Return patient display list	11
RPR	Return patient list	11
RQA	Request patient authorization	11
RQC	Request clinical information	11
RQI	Request patient information	11
RQP	Request patient demographics	11
RQQ	Event replay query	2
RRA	Pharmacy/treatment administration acknowledgment message	4
RRD	Pharmacy/treatment dispense acknowledgment message	4
RRE	Pharmacy/treatment encoded order acknowledgment message	4
RRG	Pharmacy/treatment give acknowledgment message	4
RRI	Return referral information	11
RSP	Segment pattern response	5
RTB	Tabular response	5
SIU	Schedule information unsolicited	10
SPQ	Stored procedure request	2
SQM	Schedule query message	10

Message	Description	Chapter
SQR	Schedule query response	10
SRM	Schedule request message	10
SRR	Scheduled request response	10
SSR	Specimen status request message	13
SSU	Specimen status update message	13
SUR	Summary product experience report	7
TBR	Tabular data response	2
TCR	Automated equipment test code settings request message	13
TCU	Automated equipment test code settings update message	13
UDM	Unsolicited display update message	2
VQQ	Virtual table query	2
VXQ	Query for vaccination record	4
VXR	Vaccination record response	4
VXU	Unsolicited vaccination record update	4
VXX	Response for vaccination query with multiple PID matches	4

2.17.2 Event type table

HL7 Table 0003 - Event type

Value	Description
A01	ADT/ACK - Admit/visit notification
A02	ADT/ACK - Transfer a patient
A03	ADT/ACK - Discharge/end visit
A04	ADT/ACK - Register a patient
A05	ADT/ACK - Pre-admit a patient
A06	ADT/ACK - Change an outpatient to an inpatient
A07	ADT/ACK - Change an inpatient to an outpatient
A08	ADT/ACK - Update patient information
A09	ADT/ACK - Patient departing - tracking
A10	ADT/ACK - Patient arriving - tracking
A11	ADT/ACK - Cancel admit/visit notification
A12	ADT/ACK - Cancel transfer
A13	ADT/ACK - Cancel discharge/end visit
A14	ADT/ACK - Pending admit
A15	ADT/ACK - Pending transfer
A16	ADT/ACK - Pending discharge
A17	ADT/ACK - Swap patients
A18	ADT/ACK - Merge patient information **(for backward compatibility only)**

Value	Description
A19	QRY/ADR - Patient query
A20	ADT/ACK - Bed status update
A21	ADT/ACK - Patient goes on a “leave of absence”
A22	ADT/ACK - Patient returns from a “leave of absence”
A23	ADT/ACK - Delete a patient record
A24	ADT/ACK - Link patient information
A25	ADT/ACK - Cancel pending discharge
A26	ADT/ACK - Cancel pending transfer
A27	ADT/ACK - Cancel pending admit
A28	ADT/ACK - Add person information
A29	ADT/ACK - Delete person information
A30	ADT/ACK - Merge person information **(for backward compatibility only)**
A31	ADT/ACK - Update person information
A32	ADT/ACK - Cancel patient arriving - tracking
A33	ADT/ACK - Cancel patient departing - tracking
A34	ADT/ACK - Merge patient information - patient ID only **(for backward compatibility only)**
A35	ADT/ACK - Merge patient information - account number only **(for backward compatibility only)**
A36	ADT/ACK - Merge patient information - patient ID and account number **(for backward compatibility only)**
A37	ADT/ACK - Unlink patient information
A38	ADT/ACK - Cancel pre-admit
A39	ADT/ACK - Merge person – patient ID **(for backward compatibility only)**
A40	ADT/ACK - Merge patient – patient identifier list
A41	ADT/ACK - Merge account - patient account number
A42	ADT/ACK - Merge visit - visit number
A43	ADT/ACK - Move patient information – patient identifier list
A44	ADT/ACK - Move account information - patient account number
A45	ADT/ACK - Move visit information - visit number
A46	ADT/ACK - Change patient ID **(for backward compatibility only)**
A47	ADT/ACK - Change patient identifier list
A48	ADT/ACK - Change alternate patient ID **(for backward compatibility only)**
A49	ADT/ACK - Change patient account number
A50	ADT/ACK - Change visit number
A51	ADT/ACK - Change alternate visit ID
A52	ADT/ACK – Cancel leave of absence for a patient
A53	ADT/ACK – Cancel patient returns from a leave of absence
A54	ADT/ACK - Change attending doctor

Value	Description
A55	ADT/ACK – Cancel change attending doctor
A60	ADT/ACK – Update allergy information
A61	ADT/ACK – Change consulting doctor
A62	ADT/ACK – Cancel change consulting doctor
B01	PMU/ACK – Add personnel record
B02	PMU/ACK – Update personnel record
B03	PMU/ACK – Delete personnel re cord
B04	PMU/ACK – Active practicing person
B05	PMU/ACK – Deactivate practicing person
B06	PMU/ACK – Terminate practicing person
C01	CRM - Register a patient on a clinical trial
C02	CRM - Cancel a patient registration on clinical trial (for clerical mistakes only)
C03	CRM - Correct/update registration information
C04	CRM - Patient has gone off a clinical trial
C05	CRM - Patient enters phase of clinical trial
C06	CRM - Cancel patient entering a phase (clerical mistake)
C07	CRM - Correct/update phase information
C08	CRM - Patient has gone off phase of clinical trial
C09	CSU - Automated time intervals for reporting, like monthly
C10	CSU - Patient completes the clinical trial
C11	CSU - Patient completes a phase of the clinical trial
C12	CSU - Update/correction of patient order/result information
I01	RQI/RPI - Request for insurance information
I02	RQI/RPL - Request/receipt of patient selection display list
I03	RQI/RPR - Request/receipt of patient selection list
I04	RQD/RPI - Request for patient demographic data
I05	RQC/RCI - Request for patient clinical information
I06	RQC/RCL - Request/receipt of clinical data listing
I07	PIN/ACK - Unsolicited insurance information
I08	RQA/RPA - Request for treatment authorization information
I09	RQA/RPA - Request for modification to an authorization
I10	RQA/RPA - Request for resubmission of an authorization
I11	RQA/RPA - Request for cancellation of an authorization
I12	REF/RRI - Patient referral
I13	REF/RRI - Modify patient referral
I14	REF/RRI - Cancel patient referral
I15	REF/RRI - Request patient referral status

Value	Description
J01	QCN/ACK – Cancel query/acknowledge message
J02	QSX/ACK – Cancel subscription/acknowledge message
K11	RSP - Segment pattern response
K13	RTB - Tabular response
K15	RDY - Display response
K21	RSP – Get person demographics response
K22	RSP – Find candidates response
K23	RSP – Get corresponding identifiers response
K24	RSP – Allocate identifiers response
K25	RSP - Personnel Information by Segment Response
M01	MFN/MFK - Master file not otherwise specified (**for backward compatibility only**)
M02	MFN/MFK - Master file – staff practitioner
M03	MFN/MFK - Master file - test/observation (**for backward compatibility only**)
M04	MFN/MFK - Master files charge description
M05	MFN/MFK - Patient location master file
M06	MFN/MFK - Clinical study with phases and schedules master file
M07	MFN/MFK - Clinical study without phases but with schedules master file
M08	MFN/MFK - Test/observation (numeric) master file
M09	MFN/MFK - Test/observation (categorical) master file
M10	MFN/MFK - Test /observation batteries master file
M11	MFN/MFK - Test/calculated observations master file
M12	MFN/MFK – Master file notification message
N01	NMQ/NMR - Application management query message
N02	NMD/ACK - Application management data message (unsolicited)
O01	ORM - Order message (also RDE, RDS, RGV, RAS)
O02	ORR - Order response (also RRE, RRD, RRG, RRA)
O03	OMD – Diet order
O04	ORD – Diet order acknowledgment
O05	OMS – Stock requisition order
O06	ORS – Stock requisition acknowledgment
O07	OMN – Non-stock requisition order
O08	ORN – Non-stock requisition acknowledgment
O09	OMP – Pharmacy/treatment order
O10	ORP – Pharmacy/treatment order acknowledgment
O11	RDE – Pharmacy/treatment encoded order
O12	RRE – Pharmacy/treatment encoded order acknowledgment
O13	RDS – Pharmacy/treatment dispense

Value	Description
O14	RRD – Pharmacy/treatment dispense acknowledgment
O15	RGV – Pharmacy/treatment give
O16	RRG – Pharmacy/treatment give acknowledgment
O17	RAS – Pharmacy/treatment administration
O18	RRA – Pharmacy/treatment administration acknowledgment
O19	OMG – General clinical order
O20	ORG/ORL – General clinical order response
O21	OML - Laboratory order
022	ORL - General laboratory order response message to any OML
P01	BAR/ACK - Add patient accounts
P02	BAR/ACK - Purge patient accounts
P03	DFT/ACK - Post detail financial transaction
P04	QRY/DSP – Generate bill and A/R statements
P05	BAR/ACK – Update account
P06	BAR/ACK - End account
P07	PEX - Unsolicited initial individual product experience report
P08	PEX - Unsolicited update individual product experience report
P09	SUR - Summary product experience report
P10	BAR/ACK –Transmit Ambulatory Payment Classification(APC)
PC1	PPR - PC/ problem add
PC2	PPR - PC/ problem update
PC3	PPR - PC/ problem delete
PC4	QRY - PC/ problem query
PC5	PRR - PC/ problem response
PC6	PGL - PC/ goal add
PC7	PGL - PC/ goal update
PC8	PGL - PC/ goal delete
PC9	QRY - PC/ goal query
PCA	PPV - PC/ goal response
PCB	PPP - PC/ pathway (problem-oriented) add
PCC	PPP - PC/ pathway (problem-oriented) update
PCD	PPP - PC/ pathway (problem-oriented) delete
PCE	QRY - PC/ pathway (problem-oriented) query
PCF	PTR - PC/ pathway (problem-oriented) query response
PCG	PPG - PC/ pathway (goal-oriented) add
PCH	PPG - PC/ pathway (goal-oriented) update
PCJ	PPG - PC/ pathway (goal-oriented) delete

Value	Description
PCK	QRY - PC/ pathway (goal-oriented) query
PCL	PPT - PC/ pathway (goal-oriented) query response
Q01	QRY/DSR - Query sent for immediate response
Q02	QRY/QCK - Query sent for deferred response
Q03	DSR/ACK - Deferred response to a query
Q04	EQQ – Embedded query language query
Q05	UDM/ACK - Unsolicited display update message
Q06	OSQ/OSR - Query for order status
Q07	VQQ – Virtual table query
Q08	SPQ – Stored procedure request
Q09	RQQ – event replay query
Q16	QSB – Create subscription
Q17	QVR – Query for previous events
Q21	QBP – Get person demographics
Q22	QBP – Find candidates
Q23	QBP – Get corresponding identifiers
Q24	QBP – Allocate identifiers
Q25	QBP - Personnel Information by Segment Query
Q26	ROR - Pharmacy/treatment order response
Q27	RAR - Pharmacy/treatment administration information
Q28	RDR - Pharmacy/treatment dispense information
Q29	RER - Pharmacy/treatment encoded order information
Q30	RGR - Pharmacy/treatment dose information
QNC	Varies - Query cancellation
R01	ORU/ACK - Unsolicited transmission of an observation message
R02	QRY - Query for results of observation
R03	QRY/DSR Display-oriented results, query/unsol. update **(for backward compatibility only) (Replaced by Q05)**
R04	ORF - Response to query; transmission of requested observation
ROR	ROR - Pharmacy prescription order query response
R07	EDR - Enhanced Display Response
R08	TBR - Tabular Data Response
R09	ERP - Event Replay Response
R21	OUL – Unsolicited laboratory observation
S01	SRM/SRR - Request new appointment booking
S02	SRM/SRR - Request appointment rescheduling
S03	SRM/SRR - Request appointment modification
S04	SRM/SRR - Request appointment cancellation

Value	Description
S05	SRM/SRR - Request appointment discontinuation
S06	SRM/SRR - Request appointment deletion
S07	SRM/SRR - Request addition of service/resource on appointment
S08	SRM/SRR - Request modification of service/resource on appointment
S09	SRM/SRR - Request cancellation of service/resource on appointment
S10	SRM/SRR - Request discontinuation of service/resource on appointment
S11	SRM/SRR - Request deletion of service/resource on appointment
S12	SIU/ACK - Notification of new appointment booking
S13	SIU/ACK - Notification of appointment rescheduling
S14	SIU/ACK - Notification of appointment modification
S15	SIU/ACK - Notification of appointment cancellation
S16	SIU/ACK - Notification of appointment discontinuation
S17	SIU/ACK - Notification of appointment deletion
S18	SIU/ACK - Notification of addition of service/resource on appointment
S19	SIU/ACK - Notification of modification of service/resource on appointment
S20	SIU/ACK - Notification of cancellation of service/resource on appointment
S21	SIU/ACK - Notification of discontinuation of service/resource on appointment
S22	SIU/ACK - Notification of deletion of service/resource on appointment
S23	SIU/ACK - Notification of blocked schedule time slot(s)
S24	SIU/ACK - Notification of opened (“unblocked”) schedule time slot(s)
S25	SQM/SQR - Schedule query message and response
S26	SIU/ACK Notification that patient did not show up for schedule appointment
T01	MDM/ACK - Original document notification
T02	MDM/ACK - Original document notification and content
T03	MDM/ACK - Document status change notification
T04	MDM/ACK - Document status change notification and content
T05	MDM/ACK - Document addendum notification
T06	MDM/ACK - Document addendum notification and content
T07	MDM/ACK - Document edit notification
T08	MDM/ACK - Document edit notification and content
T09	MDM/ACK - Document replacement notification
T10	MDM/ACK - Document replacement notification and content
T11	MDM/ACK - Document cancel notification
T12	QRY/DOC - Document query
U01	ESU/ACK – Automated equipment status update
U02	ESR/ACK – Automated equipment status request
U03	SSU/ACK - Specimen status update

Value	Description
U04	SSR/ACK - specimen status request
U05	INU/ACK - Automated equipment inventory update
U06	INR/ACK – Automated equipment inventory request
U07	EAC/ACK – Automated equipment command
U08	EAR/ACK – Automated equipment response
U09	EAN/ACK – Automated equipment notification
U10	TCU/ACK – Automated equipment test code settings update
U11	TCR/ACK – Automated equipment test code settings request
U12	LSU/ACK – Automated equipment log/service update
U13	LSR/ACK – Automated equipment log/service request
V01	VXQ - Query for vaccination record
V02	VXX - Response to vaccination query returning multiple PID matches
V03	VXR - Vaccination record response
V04	VXU - Unsolicited vaccination record update
Varies	MFQ/MFR - Master files query (use event same as asking for e.g., M05 - location)
W01	ORU - Waveform result, unsolicited transmission of requested information
W02	QRF - Waveform result, response to query

2.17.3 Message structure table

HL7 Table 0354 - Message structure

Value	Events
ACK	Varies
ADR_A19	A19
ADT_A01	A01, A04, A08, A13
ADT_A02	A02
ADT_A03	A03
ADT_A05	A05, A14, A28, A31
ADT_A06	A06, A07
ADT_A09	A09, A10, A11, A12
ADT_A15	A15
ADT_A16	A16
ADT_A17	A17
ADT_A18	A18
ADT_A20	A20
ADT_A21	A21, A22, A23, A25, A26, A27, A29, A32, A33
ADT_A24	A24

Value	Events
ADT_A30	A30, A34, A35, A36, A46, A47, A48, A49
ADT_A37	A37
ADT_A38	A38
ADT_A39	A39, A40, A41, A42
ADT_A43	A43, A44
ADT_A45	A45
ADT_A50	A50, A51
ADT_A52	A52, A53, A55
ADT_A54	A54
ADT_A60	A60
ADT_A61	A61, A62
BAR_P01	P01
BAR_P02	P02
BAR_P05	P05
BAR_P06	P06
BAR_P10	P10
CRM_C01	C01, C02, C03, C04, C05, C06, C07, C08
CSU_C09	C09, C10, C11, C12
DFT_P03	P03
DOC_T12	T12
DSR_P04	P04
DSR_Q01	Q01
DSR_Q03	Q03
EAC_U07	U07
EAN_U09	U09
EAR_U08	U08
EDR_R07	R07
EQQ_Q04	Q04
ERP_R09	R09
ESR_U02	U02
ESR_U02	U02
ESU_U01	U01
INR_U06	U06
INU_U05	U05
LSU_U12	U12, U13
MDM_T01	T01, T03, T05, T07, T09, T11
MDM_T02	T02, T04, T06, T08, T10

Value	Events
MFD_MFA	MFA
MFK_M01	M01, M02, M03, M04, M05, M06, M07, M08, M09, M10, M11
MFN_M01	M01
MFN_M02	M02
MFN_M03	M03
MFN_M04	M04
MFN_M05	M05
MFN_M06	M06
MFN_M07	M07
MFN_M08	M08
MFN_M09	M09
MFN_M10	M10
MFN_M11	M11
MFN_M12	M12
MFQ_M01	M01, M02, M03, M04, M05, M06
MFR_M01	M01, M02, M03, M04, M05, M06
NMD_N02	N02
NMQ_N01	N01
NMR_N01	N01
OMD_O03	O03
OMG_O19	O19
OML_O21	O21
OMN_O07	007
OMP_O09	O09
OMS_O05	O05
ORD_O04	O04
ORF_R04	R04
ORG_O20	O20
ORL_O22	022
ORM_O01	O01
ORN_008	O08
ORP_O10	O10
ORR_O02	O02
ORR_O02	O02
ORS_O06	O06
ORU_R01	R01
OSQ_Q06	Q06

Value	Events
OSR_Q06	Q06
OUL_R21	R21
PEX_P07	P07, P08
PGL_PC6	PC6, PC7, PC8
PMU_B01	B01, B02
PMU_B03	B03
PMU_B04	B04, B05
PPG_PCG	PCC, PCG, PCH, PCJ
PPP_PCB	PCB, PCD
PPR_PC1	PC1, PC2, PC3
PPT_PCL	PCL
PPV_PCA	PCA
PRR_PC5	PC5
PTR_PCF	PCF
QBP_Q11	Q11
QBP_Q13	Q13
QBP_Q15	Q15
QBP_Q21	Q21, Q22, Q23,Q24, Q25
QCK_Q02	Q02
QCN_J01	J01, J02
QRY_A19	A19
QRY_P04	P04
QRY_PC4	PC4, PC9, PCE, PCK
QRY_Q01	Q01
QRY_Q02	Q02
QRY_Q26	Q26
QRY_Q27	Q27
QRY_Q28	Q28
QRY_Q29	Q29
QRY_Q30	Q30
QRY_R02	R02
QRY_T12	T12
QSB_Q16	Q16
QVR_Q17	Q17
RAS_O17	O17
RCI_I05	I05
RCL_I06	I06

Value	Events
RDE_O01	O01
RDR_RDR	RDR
RDS_O13	O13
RDY_K15	K15
REF_I12	I12, I13, I14, I15
RER_RER	RER
RGR_RGR	RGR
RGV_O15	O15
ROR_ROR	ROR
RPA_I08	I08, I09. I10, 1II
RPI_I0I	I01, I04
RPL_I02	I02
RPR_I03	I03
RQA_I08	I08, I09, I10, I11
RQC_I05	I05, I06
RQI_I0I	I01, I02, I03, I07
RQP_I04	I04
RQQ_Q09	Q09
RRA_O02	O02
RRA_O18	O18
RRD_O14	O14
RRE_O12	O12
RRG_O16	O16
RRI_I12	I12, I13, I14, I15
RSP_K11	K11
RSP_K21	K21
RSP_K22	K22
RSP_K23	K23, K24
RTB_K13	K13
SPQ_Q08	Q08
SQM_S25	S25
SQR_S25	S25
SRM_S01	S01, S02, S03, S04, S05, S06, S07, S08, S09, S10, S11
SRR_S01	S01, S02, S03, S04, S05, S06, S07, S08, S09, S10, S11
SSR_U04	U04
SSU_U03	U03
SUR_P09	P09

Value	Events
SUR_P09	P09
TBR_R08	R08
TBR_R09	R09
TCU_U10	U10, U11
UDM_Q05	Q05
VQQ_Q07	Q07
VXQ_V01	V01
VXR_V03	V03
VXU_V04	V04
VXX_V02	V02
ORU_W01	W01
QRF_W02	W02

2.17.4 Data types table

The LEN column in the data types table specifies the maximum length where there is an agreed upon specification across chapters

HL7 Table 0440 – Data types

Value	Description	LEN	Table#	Comment	HL7 Section Reference
AD	Address		0399/0190	Replaced by XAD as of v 2.3	2.9.1
CD	Channel definition			For waveform data only; See Chapter 7, Section 7.16.2 for full specifications.	2.9.2
CE	Coded element	250	0396		2.9.3
CF	Coded element with formatted values		0396		2.9.4
CK	Composite ID with check digit	250	0061/0363		2.9.5
CM	Composite			Deprecated in v 2.3; retained for backward compatibility only.	2.9.6
CN	Composite ID number and name	250	0360/0297/0363	Replaced by XCN as of v 2.3	2.9.7
CNE	Coded with no exceptions	250	0396		2.9.8
CP	Composite price		0205/0298	Replaces MO as of v 2.3.	2.9.9
CQ	Composite quantity with units			In future versions, CQ fields should be avoided because the same data can usually be sent as two separate fields, one with the value and one with the units as a CE data	2.9.10

Value	Description	LEN	Table#	Comment	HL7 Section Reference
				type.	
CWE	Coded with exceptions	250	0396		2.9.11
CX	Extended composite ID with check digit	250	0061/0363/ 0203/		2.9.12
DLN	Driver's license number		0333		2.9.13
DR	Date/time range				2.9.14
DT	Date				2.9.15
ED	Encapsulated data		0191/0291/ 0299	Supports ASCII MIME-encoding of binary data.	2.9.16
EI	Entity identifier		0363/0301		2.9.17
FC	Financial class		0064		2.9.18
FN	Family name			Appears ONLY in the PN and other PN-containing data types (PPN, XCN, XPN).	2.9.19
FT	Formatted text	65536			2.9.20
HD	Hierarchic designator		0300/0301		2.9.21
ID	Coded values for HL7 tables				2.9.22
IS	Coded value for user-defined tables				2.9.23
JCC	Job code/class		0327/0328		2.9.24
MA	Multiplexed array			For waveform data only, see Chapter 7, Section 7.14.1.2.	2.9.25
MO	Money			Intent is that it appear only as a component of data type CP. Used independently in chapter 8, section 8.10.3.	2.9.26
NA	Numeric array			For waveform data only, see Chapter 7, Section 7.14.1.1.	2.9.27
NM	Numeric				2.9.28
PL	Person location		0302/0303/ 0304/0306/ 0305/0307/ 0308		2.9.29
PN	Person name	48	0360	Replaced by XPN as of v 2.3	2.9.30
PPN	Performing person time stamp	250	0360/0297/ 0363/0200/ 0061/0465/ 0448/0444	equivalent of an XCN joined with a TS	2.9.31
PT	Processing type		0103/0207		2.9.32
QIP	Query input parameter list				2.9.33
QSC	Query selection criteria		0209/0210		2.9.34
RCD	Row column definition		0440		2.9.35

Value	Description	LEN	Table#	Comment	HL7 Section Reference
RI	Repeat interval		0335	For scheduling data only. See Chapter 10	2.9.36
RP	Reference pointer		0191/0291		2.9.37
SAD	Street Address			Appears ONLY in the XAD data type.	2.9.38
SCV	Scheduling class value pair			For scheduling data only. See Chapter 10	2.9.39
SI	Sequence ID				2.9.40
SN	Structured numeric				2.9.41
SRT	Sort order		0397		2.9.42
ST	String	199			2.9.43
TM	Time				2.9.44
TN	Telephone number			Replaced by XTN as of v 2.3	2.9.45
TQ	Timing/quantity			For detailed specifications see Chapter 4, Section 4.3.	2.9.46
TS	Time stamp				2.9.47
TX	Text data	65536			2.9.48
VH	Visiting hours		0267		2.9.49
VID	Version identifier		0104		2.9.50
XAD	Extended address	250	0399/0190/ 0289/0288/ 0465	Replaces AD as of v 2.3	2.9.51
XCN	Extended composite ID number and name	250	360/297/36 3/200/61/2 03/465/448 /444	Replaces CN as of v 2.3	2.9.52
XON	Extended composite name and ID number for organizations	250	0204/0061/ 0363/0203/ 0465		2.9.53
XPN	Extended person name	250	0360/0200/ 0465/0448/ 0444	Replaces PN as of v 2.3.	2.9.54
XTN	Extended telecommunica- tions number	250	0201/0202	Replaces TN as of v 2.3	2.9.55

2.17.5 Coding system table

Chapter 7 is the steward for the Coding Systems table. The table is copied here for reader convenience.

User-defined Table 0396 - Coding System

Value	Description	Comment / Source	Category
99zzz or L	Local general code (where z is an alphanumeric character)	Locally defined codes for purpose of sender or receiver. Local codes can be identified by L (for backward compatibility) or 99zzz (where z is an alphanumeric character).	General code
ACR	American College of Radiology finding codes	Index for Radiological Diagnosis Revised, 3rd Edition 1986, American College of Radiology, Reston, VA.	Specific Non-Drug Code
ART	WHO Adverse Reaction Terms	WHO Collaborating Centre for International Drug Monitoring, Box 26, S-751 03, Uppsala, Sweden.	Drug code
AS4	ASTM E1238/ E1467 Universal	American Society for Testing & Materials and CPT4 (see Appendix X1 of Specification E1238 and Appendix X2 of Specification E1467).	Specific Non-Drug Code
AS4E	AS4 Neurophysiology Codes	ASTM's diagnostic codes and test result coding/grading systems for clinical neurophysiology. See ASTM Specification E1467, Appendix 2.	Specific Non-Drug Code
ATC	American Type Culture Collection	Reference cultures (microorganisms, tissue cultures, etc.), related biological materials and associated data. American Type Culture Collection, 12301 Parklawn Dr, Rockville MD, 20852. (301) 881-2600. http://www.atcc.org	Specific Non-Drug Code
C4	CPT-4	American Medical Association, P.O. Box 10946, Chicago IL 60610.	Specific Non-Drug Code
C5	CPT-5	(under development – same contact as above)	Specific Non-Drug Code
CAS	Chemical abstract codes	These include unique codes for each unique chemical, including all generic drugs. The codes do not distinguish among different dosing forms. When multiple equivalent CAS numbers exist, use the first one listed in USAN. USAN 1990 and the USP dictionary of drug names, William M. Heller, Ph.D., Executive Editor, United States Pharmacopeial Convention, Inc., 12601 Twinbrook Parkway, Rockville, MD 20852.	Drug code
CD2	CDT-2 Codes	American Dental Association's Current Dental Terminology (CDT-2) code. American Dental Association, 211 E. Chicago Avenue,. Chicago, Illinois 60611.	Specific Non-Drug Code
CDCA	CDC Analyte Codes	As above, for CDCM	
CDCM	CDC Meth-ods/Instruments Codes	Public Health Practice Program Office, Centers for Disease Control and Prevention, 4770 Buford Highway, Atlanta, GA, 30421. Also available via FTP: ftp.cdc.gov/pub/laboratory_info/CLIA and Gopher: gopher.cdc.gov:70/11/laboratory_info/CLIA	Drug code
CDS	CDC Surveillance	CDC Surveillance Codes. For data unique to specific public health surveillance requirements. Epidemiology Program Office, Centers for Disease Control and Prevention, 1600 Clifton Rd, Atlanta, GA, 30333. (404) 639-3661.	Specific Non-Drug Code
CE	CEN ECG diagnostic codes	CEN PT007. A quite comprehensive set of ECG diagnostic codes (abbreviations) and descriptions published as a pre-standard by CEN TC251. Available from CEN TC251 secretariat, c/o Georges DeMoor, State University Hospital Gent, De Pintelaan 185-5K3, 9000 Gent, Belgium or Jos Willems, University of Gathuisberg, 49 Herestraat, 3000 Leuven, Belgium.	Specific Non-Drug Code
CLP	CLIP	Simon Leeming, Beth Israel Hospital, Boston MA. Codes for radiology reports.	Specific Non-Drug Code
CPTM	CPT Modifier Code	Available for the AMA at the address listed for CPT above. These codes are found in Appendix A of CPT 2000 Standard Edition. (CPT 2000 Standard Edition, American Medical	Specific Non-Drug Code

		Association, Chicago, IL).	
CST	COSTART	International coding system for adverse drug reactions. In the USA, maintained by the FDA, Rockville, MD.	Drug code
CVX	CDC Vaccine Codes	National Immunization Program, Centers for Disease Control and Prevention, 1660 Clifton Road, Atlanta, GA, 30333	Drug code
DCL	DICOM Class Label	From the Message Standards Classes table of the SNOMED-DICOM-Microglossary. College of American Pathologists, Skokie, IL, 60077-1034	Specific Non-Drug Code
DCM	DICOM modality codes	Dean Bidgood, MD; Duke University Medical Center, Durham NC. Digital Imaging and Communications in Medicine (DICOM). From NEMA Publications PS-3.1 – PS 3.12: The ACR-NEMA DICOM Standard. National Electrical Manufacturers Association (NEMA). Rosslyn, VA, 22209., 1992, 1993, 1995	Specific Non-Drug Code
DQL	DICOM Query Label	HL7 Image Management Special Interest Group, Health Level Seven, Ann Arbor, MI.	Specific Non-Drug Code
E	EUCLIDES	Available from Euclides Foundation International nv, Excelsiorlaan 4A, B-1930 Zaventem, Belgium; Phone: 32 2 720 90 60.	Specific Non-Drug Code
E5	Euclides quantity codes	Available from Euclides Foundation International nv (see above)	Specific Non-Drug Code
E6	Euclides Lab method codes	Available from Euclides Foundation International nv, Excelsiorlaan 4A, B-1930 Zaventem, Belgium; Phone: 32 2 720 90 60.	Specific Non-Drug Code
E7	Euclides Lab equipment codes	Available from Euclides Foundation International nv (see above)	Specific Non-Drug Code
ENZC	Enzyme Codes	Enzyme Committee of the International Union of Biochemistry and Molecular Biology. Enzyme Nomenclature: Recommendations on the Nomenclature and Classification of Enzyme-Catalysed Reactions. London: Academic Press, 1992.	Specific Non-Drug Code
FDDC	First DataBank Drug Codes	National Drug Data File. Proprietary product of First DataBank, Inc. (800) 633-3453, or http://www.firstdatabank.com.	Drug code
FDDX	First DataBank Diagnostic Codes	Used for drug-diagnosis interaction checking. Proprietary product of First DataBank, Inc. As above for FDDC.	Drug code
FDK	FDA K10	Dept. of Health & Human Services, Food & Drug Administration, Rockville, MD 20857. (device & analyte process codes).	Specific Non-Drug Code
HB	HIBCC	Health Industry Business Communications Council, 5110 N. 40th St., Ste 120, Phoenix, AZ 85018.	Specific Non-Drug Code
HCPCS	HCFA Common Procedure Coding System	HCPCS: contains codes for medical equipment, injectable drugs, transportation services, and other services not found in CPT4.	Specific Non-Drug Code
HHC	Home Health Care	Home Health Care Classification System; Virginia Saba, EdD, RN; Georgetown University School of Nursing; Washington, DC.	Specific Non-Drug Code
HI	Health Outcomes	Health Outcomes Institute codes for outcome variables available (with responses) from Stratis Health (formerly Foundation for Health Care Evaluation and Health Outcomes Institute), 2901 Metro Drive, Suite 400, Bloomington, MN, 55425-1525; (612) 854-3306 (voice); (612) 853-8503 (fax); dziegen@winternet.com. See examples in the Implementation Guide.	Specific Non-Drug Code
HL7nnnn	HL7 Defined Codes where nnnn is the HL7	Health Level Seven where nnnn is the HL7 table number	General code

	table number		
HPC	HCFA Procedure Codes (HCPCS)	Health Care Financing Administration (HCFA) Common Procedure Coding System (HCPCS) including modifiers.[4]	Specific Non-Drug Code
I10	ICD-10	World Health Publications, Albany, NY.	Specific Non-Drug Code
I10P	ICD-10 Procedure Codes	Procedure Coding System (ICD-10-PCS.) See http://www/hcfa.gov/stats/icd10.icd10.htm for more information.	Specific Non-Drug Code
I9	ICD9	World Health Publications, Albany, NY.	Specific Non-Drug Code
I9C	ICD-9CM	Commission on Professional and Hospital Activities, 1968 Green Road, Ann Arbor, MI 48105 (includes all procedures and diagnostic tests).	Specific Non-Drug Code
IBT	ISBT	International Society of Blood Transfusion. Blood Group Terminology 1990. VOX Sanquines 1990 58(2):152-169.	Specific Non-Drug Code
IC2	ICHPPC-2	International Classification of Health Problems in Primary Care, Classification Committee of World Organization of National Colleges, Academies and Academic Associations of General Practitioners (WONCA), 3rd edition. An adaptation of ICD9 intended for use in General Medicine, Oxford University Press.	Specific Non-Drug Code
ICDO	International Classification of Diseases for Oncology	International Classification of Diseases for Oncology, 2nd Edition. World Health Organization: Geneva, Switzerland, 1990. Order from: College of American Pathologists, 325 Waukegan Road, Northfield, IL, 60093-2750. (847) 446-8800.	Specific Non-Drug Code
ICS	ICCS	Commission on Professional and Hospital Activities, 1968 Green Road, Ann Arbor, MI 48105.	Specific Non-Drug Code
ICSD	International Classification of Sleep Disorders	International Classification of Sleep Disorders Diagnostic and Coding Manual, 1990, available from American Sleep Disorders Association, 604 Second Street SW, Rochester, MN 55902	Specific Non-Drug Code
ISOnnnn	ISO Defined Codes where nnnn is the ISO table number	International Standards Organization where nnnn is the ISO table number	General code
IUPP	IUPAC/IFCC Property Codes	International Union of Pure and Applied Chemistry/International Federation of Clinical Chemistry. The Silver Book: Compendium of terminology and nomenclature of properties in clinical laboratory sciences. Oxford: Blackwell Scientific Publishers, 1995. Henrik Olesen, M.D., D.M.Sc., Chairperson, Department of Clinical Chemistry, KK76.4.2, Rigshospitalet, University Hospital of Copenhagen, DK-2200, Copenhagen. http://inet.uni-c.dk/~qukb7642/	Specific Non-Drug Code
IUPC	IUPAC/IFCC Component Codes	Codes used by IUPAC/IFF to identify the component (analyte) measured. Contact Henrik Olesen, as above for IUPP.	Specific Non-Drug Code

4 The HCPCS code is divided into three "levels." Level I includes the entire CPT-4 code by reference. Level II includes the American Dental Association's Current Dental Terminology (CDT-2) code by reference. Level II also includes the genuine HCPCS codes, approved and maintained jointly by the Alpha-Numeric Editorial Panel, consisting of HCFA, the Health Insurance Association of America, and the Blue Cross and Blue Shield Association. Level III are codes developed locally by Medicare carriers. The HCPCS modifiers are divided into the same three levels, I being CPT-4 modifiers, II CDT-2 and genuine HCPCS modifiers, and III being locally agreed modifiers.

The genuine HCPCS codes and modifiers of level II can be found at http://www.hcfa.gov/stats/anhcpcdl.htm. HCFA distributes the HCPCS codes via the National Technical Information Service (NTIS, www.ntis.gov) and NTIS distribution includes the CDT-2 part of HCPCS Level II, but does not include the CPT-4 part (Level I). HCFA may distribute the CPT-4 part to its contractors.

JC8	Japanese Chemistry	Clinical examination classification code. Japan Association of Clinical Pathology. Version 8, 1990. A multiaxial code including a subject code (e.g., Rubella = 5f395, identification code (e.g., virus ab IGG), a specimen code (e.g., serum =023) and a method code (e.g., ELISA = 022)	Specific Non-Drug Code
LB	Local billing code	Local billing codes/names (with extensions if needed).	General code
LN	Logical Observation Identifier Names and Codes (LOINC®)	Regenstrief Institute, c/o LOINC, 1050 Wishard Blvd., 5th floor, Indianapolis, IN 46202. 317/630-7433. Available from the Regenstrief Institute server at http://www. Regenstrief.org/loinc/loinc.htm. Also available via HL7 file server: FTP/Gopher (www.mcis.duke.edu/standards/ termcode/loinclab and www.mcis.duke.edu/standards/termcode/loinclin) and World Wide Web (http:// www.mcis.duke.edu/ standards/termcode/loincl.htm). January 2000 version has identifiers, synonyms and cross-reference codes for reporting over 26,000 laboratory and related observations and 1,500 clinical measures.	Specific Non-Drug Code
MCD	Medicaid	Medicaid billing codes/names.	Specific Non-Drug Code
MCR	Medicare	Medicare billing codes/names.	Specific Non-Drug Code
MDDX	Medispan Diagnostic Codes	Codes Used for drug-diagnosis interaction checking. Proprietary product. Hierarchical drug codes for identifying drugs down to manufacturer and pill size. MediSpan, Inc., 8425 Woodfield Crossing Boulevard, Indianapolis, IN 46240. Tel: (800) 428-4495. WWW: http://www.espan.com/medispan/pages/medhome.html. As above for MGPI.	Drug code
MEDC	Medical Economics Drug Codes	Proprietary Codes for identifying drugs. Proprietary product of Medical Economics Data, Inc. (800) 223-0581.	Drug code
MEDR	Medical Dictionary for Drug Regulatory Affairs (MEDDRA)	Dr. Louise Wood, Medicines Control Agency, Market Towers, 1 Nine Elms Lane, London SW85NQ, UK Tel: (44)0 171-273-0000 WWW: http://www.open.gov.uk/mca/mcahome.htm	Drug code
MEDX	Medical Economics Diagnostic Codes	Used for drug-diagnosis interaction checking. Proprietary product of Medical Economics Data, Inc. (800) 223-0581.	Drug code
MGPI	Medispan GPI	Medispan hierarchical drug codes for identifying drugs down to manufacturer and pill size. Proprietary product of MediSpan, Inc., 8425 Woodfield Crossing Boulevard, Indianapolis, IN 46240. Tel: (800) 428-4495.	Drug code
MVX	CDC Vaccine Manufacturer Codes	As above, for CVX	Drug code
NDA	NANDA	North American Nursing Diagnosis Association, Philadelphia, PA.	Specific Non-Drug Code
NDC	National drug codes	These provide unique codes for each distinct drug, dosing form, manufacturer, and packaging. (Available from the National Drug Code Directory, FDA, Rockville, MD, and other sources.)	Drug code
NIC	Nursing Interventions Classification	Iowa Intervention Project, College of Nursing, University of Iowa, Iowa City, Iowa	Specific Non-Drug Code
NPI	National Provider Identifier	Health Care Finance Administration, US Dep't. of Health and Human Services, 7500 Security Blvd., Baltimore, MD 21244.	Specific Non-Drug Code
OHA	Omaha System	Omaha Visiting Nurse Association, Omaha, NB.	Specific Non-Drug Code
OHA	Omaha	Omaha Visiting Nurse Association, Omaha, NB.	Specific Non-Drug Code
POS	POS Codes	HCFA Place of Service Codes for Professional Claims (see http://www.hcfa.gov/medicare/poscode.htm).	Specific Non-Drug Code

RC	Read Classification	The Read Clinical Classification of Medicine, Park View Surgery, 26 Leicester Rd., Loughborough LE11 2AG (includes drug procedure and other codes, as well as diagnostic codes).	Specific Non-Drug Code
SDM	SNOMED- DICOM Microglossary	College of American Pathologists, Skokie, IL, 60077-1034. (formerly designated as 99SDM).	Specific Non-Drug Code
SNM	Systemized Nomenclature of Medicine (SNOMED)	Systemized Nomenclature of Medicine, 2nd Edition 1984 Vols 1, 2, College of American Pathologists, Skokie, IL.	Specific Non-Drug Code
SNM3	SNOMED International	SNOMED International, 1993 Vols 1-4, College of American Pathologists, Skokie, IL, 60077-1034..	Specific Non-Drug Code
SNT	SNOMED topology codes (anatomic sites)	College of American Pathologists, 5202 Old Orchard Road, Skokie, IL 60077-1034.	Specific Non-Drug Code
UC	UCDS	Uniform Clinical Data Systems. Ms. Michael McMullan, Office of Peer Review Health Care Finance Administration, The Meadows East Bldg., 6325 Security Blvd., Baltimore, MD 21207; (301) 966 6851.	Specific Non-Drug Code
UMD	MDNS	Universal Medical Device Nomenclature System. ECRI, 5200 Butler Pike, Plymouth Meeting, PA 19462 USA. Phone: 215-825-6000, Fax: 215-834-1275.	Device code
UML	Unified Medical Language	National Library of Medicine, 8600 Rockville Pike, Bethesda, MD 20894.	Specific Non-Drug Code
UPC	Universal Product Code	The Uniform Code Council. 8163 Old Yankee Road, Suite J, Dayton, OH 45458; (513) 435 3070	Specific Non-Drug Code
UPIN	UPIN	Medicare/HCFA's universal physician identification numbers, available from Health Care Financing Administration, U.S. Dept. of Health and Human Services, Bureau of Program Operations, 6325 Security Blvd., Meadows East Bldg., Room 300, Baltimore, MD 21207	Specific Non-Drug Code
W1	WHO record # drug codes (6 digit)	World Health organization record number code. A unique sequential number is assigned to each unique single component drug and to each multi-component drug. Eight digits are allotted to each such code, six to identify the active agent, and 2 to identify the salt, of single content drugs. Six digits are assigned to each unique combination of drugs in a dispensing unit. The six digit code is identified by W1, the 8 digit code by W2.	Drug code
W2	WHO record # drug codes (8 digit)	World Health organization record number code. A unique sequential number is assigned to each unique single component drug and to each multi-component drug. Eight digits are allotted to each such code, six to identify the active agent, and 2 to identify the salt, of single content drugs. Six digits are assigned to each unique combination of drugs in a dispensing unit. The six digit code is identified by W1, the 8 digit code by W2.	Drug code
W4	WHO record # code with ASTM extension	With ASTM extensions (see Implementation Guide), the WHO codes can be used to report serum (and other) levels, patient compliance with drug usage instructions, average daily doses and more (see Appendix X1 the Implementation Guide).	Drug code
WC	WHO ATC	WHO's ATC codes provide a hierarchical classification of drugs by therapeutic class. They are linked to the record number codes listed above.	Drug code

2.17.6 Yes/no indicator table

The actual interpretation of Yes/No is context sensitive. Individual chapters will further refine the meaning of Yes/No in their specific context.

HL7 Table 0136 - Yes/no indicator

Value	Description
Y	Yes
N	No

2.18 SAMPLE CONTROL MESSAGES

2.18.1 General acknowledgment

LAB acknowledges the message that ADT sent identified as ZZ9380. (LAB and ADT, the sending and receiving system IDs, are site-defined.) Both systems are associated with the same FACILITY, 767543. The **AA** code in the MSA segment indicates that the message was accepted by the application.

```
MSH|^~\&|LAB|767543|ADT|767543|19900314130405||ACK^^ACK_ACK|XX3657|P|2.4<cr>
MSA|AA|ZZ9380<cr>
```

2.18.2 Error return

The **AR** code in MSA indicates that the application rejected the message for functional reasons. The optional ERR segment includes here that the 16th field of the PID segment with the SET ID value of 1 had an error which was defined by the locally-established code X3L. The optional text message UNKNOWN COUNTY CODE in the link is designed to help programmers and support personnel while reviewing message logs.

```
MSH|^~\&|LAB|767543|ADT|767543|199003141304-0500|| ACK^^ACK_ACK |XX3657|P|2.4<cr>
MSA|AR|ZZ9380|UNKNOWN COUNTY CODE<cr>
ERR|PID^1^16^X3L<cr>
```

2.18.3 Sequence number: initial message

The sender initiates the link with a message that has no functional content. The sequence number is 0. The message type and event code are not used.

```
MSH|^~\&|ADT|767543|LAB|767543|199003141304-0500||^|XX3657|P|2.4|0<cr>
```

The responder uses a general acknowledgment. The expected sequence number is 1.

```
MSH|^~\&|LAB|767543|ADT|767543|199003141304-0500|| ACK^^ACK_ACK |ZZ9380|P|2.4<cr>
MSA|AA|XX3657||1<cr>
```

2.18.4 Example of message fragmentation

This summarizes the methodology for splitting a single logical HL7 message among two or more actual HL7 messages. The actual specifications for this, the segment definitions of the ADD and DSC segments, and examples are in Section 2.14.2, "Continuation messages and segments".

Continuing of messages is a generic methodology that can be used for all HL7 message types. It can be used to split based on segment boundaries, on field boundaries, and to split a single field among several messages. It utilizes two specific segments, ADD and DSC, as well as a field in the message header, *MSH-14-Continuation pointer*.

When a message is continued, a unique continuation value is used. This same value will appear in MSH-14 and DSC-1 as appropriate for a single pair of messages. This allows messages to be "chained together".

Here are two examples of ways to create continuation pointers for fragmented messages. The only absolute requirement is that when the sending application values the continuation pointer, the receiving application can appropriately reconstruct the message.

Sitecode-interfaceapplicationcode-date-sequentialcounterwithindate

This will guarantee uniqueness of this field.

e.g. BWH-LDS-19990331-27 for the 27th large message to be created on March 31, within the Discharge Summary interfaces at BWH

An alternative method of valuing the continuation pointer:

Sitecode-interfaceapplicationcode-medicalrecordnumber-datetime

e.g. MGH-PCIS-1234567-19980331121314 for a message created on March 31, at 12:13:14pm for patient medical record number 1234567, within the PCIS interfaces at MGH

Sending Application Note: In the ADD segment, a trailing field delimiter, i.e. the vertical bar character, after the final field, has explicit meaning. The sending application should not include a trailing field delimiter for the last field in the ADD segment unless it has completely valued the entire field from the message being continued.

Receiving Application Note: The receiving application will need to be concerned with a single segment and a single field being continued.

Receiving a message with an empty ADD segment followed by a DSC segment is the notification that the segment preceding the ADD is being continued in a subsequent message. Note that the continuing message may not be the next one received! The receiver must match up the continuation pointer value from MSH-14 of subsequent messages to the DSC-1 continuation pointer value of the prior message. Also if the continuing message contains an ADD segment, the receiver should continue appending to the fields from the segment being continued with values from the ADD segment. For example, if OBX-5 is being continued, the continuation will appear in ADD-1 of the continuing message. If there were a value for OBX-13 of the original message, that would appear in ADD-9 of the continuing message, assuming that the remainder of the OBX segment fit into the single ADD segment.

Question: if continuing a message after the completion of a complete segment, should the continuing message have an empty ADD segment or not? Answer: No. This means that a continuing message need not have an ADD segment, if the continued message was split on a segment boundary.

Notation conventions: items within angle brackets are comments and not intended to represent a portion of an actual message. For example, <this is a comment>.

Note the multiple continuation pointer values, one for each pair of physical messages.

Message 1

```
MSH|…|<field-13>||…
PID|…
ORC|…
OBR|…
OBX|1|FT|^Discharge Summary|1|This is the first sentence of a long
message.  This is the second sentence of a long message.
```

<snip>

This is the 967th sentence of “

ADD|

DSC|**BWH-LDS-19990405-6**|

Message 2

MSH|…|<field-13>|**BWH-LDS-19990405-6**|

ADD|a long message. This is the 968th sentence of a long message.

<snip>

This is the 1001st line of

<there should be no trailing field delimiter after the last field in this ADD segment>

DSC|**BWH-LDS-19990405-7**|

Message 3

MSH|…|<field-13>|**BWH-LDS-19990405-7**|

ADD|a long message. This is the 1002nd sentence of a long message. <snip> This is the final sentence of this long message!|||||F||199707211325|

DG1|…

<end of message>

The following examples discuss an unsolicited transmission of an observation message, ORU^R01.

The expected result values in OBX-5, Observation Value, for reports (e.g. autopsy, pathology) may exceed the message length restrictions of one or more interfaces.

Thus the OBX-5, Observation Value data element will be split into more than one message.

Here’s an example intended to illustrate the interpretation of Chapter 2 and 7. It reflects a single logical message broken up into three distinct messages.

Example 1, a single field being split across three messages

Message #1: ----------

Note: MSH-14, continuation pointer, is empty.

MSH|…|<field-13>||…

PID|…

ORC|…

OBR|…

OBX|1|FT|^Discharge Summary|1|This is the first sentence of a long

message. This is the second sentence of a long message.

<snip>

This is the 967th sentence of “

ADD|

DSC|<continuation-pointer-value-1>|F

Message #2: ----------

Note: MSH-14, continuation pointer, is valued with the same value as in DSC-1, continuation pointer from the message this is continuing, in this case Message #1.

```
MSH|…|<field-13>|<continuation-pointer-value-1>|
ADD|a long message.  This is the 968th sentence of a long message.
<snip>
This is the 1001st line of
<there should be no trailing field delimiter after the last field in this ADD
    segment>
DSC|<continuation-pointer-value-2>|F

Message #3: ----------------------------------------------------------------
Note: MSH-14, continuation pointer, is valued with the same value as in DSC-1,
    continuation pointer from the message this is continuing, in this case Message
    #1.

MSH|…|<field-13>|<continuation-pointer-value-2>|
ADD|a long message.  This is the 1002nd sentence of a long message. <snip> This is the
    final sentence of this long message!|||||F||199707211325|

<remaining segments after the big OBX from the original message go here, after the
    ADD segment>

PR1|…
DG1|…
```

Example 2, a single message being split across two messages, but on segment boundaries

Message #1: ----------

Note:	MSH-14, continuation pointer, is empty.

```
MSH|…|<field-13>||…
PID|…
ORC|…
OBR|…
OBX|1|FT|^Discharge Summary|1|This is the first sentence of a long
message. This is the final sentence of this long discharge
    summary!|||||F||199707211325|
DSC|<continuation-pointer-value-3>|F
```

Message #2: ----------

Note:	MSH-14, continuation pointer, is valued with the same value as in DSC-1, continuation pointer from the message this is continuing, in this case Message #1.

Note that no ADD segment is necessary, since a segment is not being split across two messages.

```
MSH|…|<field-13>|<continuation-pointer-value-3>|
PR1|…
DG1|…
```

2.18.5 Master file update examples: with original and enhanced acknowledgment protocol

This example shows the lab system using the Master Files specification to send two update test dictionary entries to an ICU system. The OM1 (observation dictionary) segment, currently under development by HL7 and ASTM, carries the dictionary information. Several varieties of acknowledgment are shown. The choice of acknowledgment mode is site-specific.

2.18.5.1 Original mode example:

```
MSH|^~\&|LABxxx|ClinLAB|ICU||19910918060544||MFN^M03|MSGID002|P|2.2
MFI|LABxxx^Lab Test Dictionary^L|UPD|||AL
MFE|MUP|199109051000|199110010000|12345^WBC^L
OM1|...
MFE|MUP|199109051015|199110010000|6789^RBC^L
OM1|...
```

Original mode acknowledgment of the HL7 message according to MFI Response Level Code of AL.

```
MSH|^~\&|ICU||LABxxx|ClinLAB|19910918060545||MFK|MSGID99002|P|2.2
MSA|AA|MSGID002
MFI|LABxxx^Lab Test Dictionary^L|UPD|||MFAA
MFA|MUP|199110010000|199110010040|S|12345^WBC^L
MFA|MUP|199110010000|199110010041|S|6789^RBC^L
```

2.18.5.2 Enhanced mode example

2.18.5.2.1 Initial message with accept acknowledgment

```
MSH|^~\&|LABxxx|ClinLAB|ICU||19910918060544||MFN^M03|MSGID002|P|2.2|||AL|AL
MFI|LABxxx^Lab Test Dictionary^L|UPD|||AL
MFE|MUP|199109051000|199110010000|12345^WBC^L
OM1|...
MFE|MUP|199109051015|199110010000|6789^RBC^L
OM1|...

MSH|^~\&|ICU||LABxxx|ClinLAB|19910918060545||MSA|MSGID99002|P|2.2
MSA|CA|MSGID002
```

2.18.5.2.2 Application acknowledgment message

```
MSH|^~\&|ICU||LABxxx|ClinLAB|19911001080504||MFK|MSGID5002|P|2.2|||AL|
MSA|AA|MSGID002
MFI|LABxxx^Lab Test Dictionary^L|UPD|||MFAA
MFA|MUP|199109051000|199110010040|S|12345^WBC^L
MFA|MUP|199109051015|199110010041|S|6789^RBC^L

MSH|^~\&|LABxxx|ClinLAB|ICU||19911001080507||ACK|MSGID444|P|2.2
MSA|CA|MSGID5002
```

2.18.5.3 Delayed application acknowledgment

Note:	If the MFN message in Section 2.18.5.2, “Enhanced mode example” had not required an application acknowledgment at the message level (i.e., the application acknowledgment code of the MSH segment = NE), the (Master Files Chapter defined) MFD message could be used to provide a delayed application level acknowledgment not tied to the original MFN message.

The following example includes an acknowledgment for an MFE segment not in the original message. This additional MFE was sent via another MFN message.

2.18.5.3.1 Initial message with accept acknowledgment

```
MSH|^~\&|LABxxx|ClinLAB|ICU||19910918060544||MFN^M03|MSGID002|P|2.2|||AL|NE
MFI|LABxxx^Lab Test Dictionary^L|UPD|||AL
MFE|MUP|199109051000|199110010000|12345^WBC^L
OM1|...
MFE|MUP|199109051015|199110010000|6789^RBC^L
OM1|...

MSH|^~\&|ICU||LABxxx|ClinLAB|19910918060545||MSA|MSGID99002|P|2.2
MSA|CA|MSGID002
```

2.18.5.3.2 Delayed application acknowledgment

```
MSH|^~\&|ICU||LABxxx|ClinLAB|19911001080504||MFD|MSGID65002|P|2.2|||AL|
MFI|LABxxx^Lab Test Dictionary^L|UPD|||MFAA
MFA|MUP|199109051000|199110010040|S|12345^WBC^L
MFA|MUP|199109051015|199110010041|S|6789^RBC^L
MFA|MUP|199109051025|199110010041|S|4339^HGB^L

MSH|^~\&|LABxxx|ClinLAB|ICU||19911001080507||ACK|MSGID444|P|2.2
MSA|CA|MSGID65002
```

2.19 OUTSTANDING ISSUES

The following items are being discussed in the Control/Query technical committee for addition to future versions of HL7:

1) Rationalization and clarification of event structures.
2) Creation of a network server for HL7 tables so that updates to them can be made public immediately, rather than waiting until the publication of the next version of the Standard.
3) Consideration of security. There are in general two types: application level security, which is partially addressed by the *security* field in the MSH segment. The second type, network security, needs to be addressed in the HL7 Implementation Guide. There are several commercially available encryption-based approaches to network level security.
4) Reviewing network application management messages for possible upgrade requirements.
5) Creation of Implementation Technology Specifications (ITSs: encoding rules equivalents) for Version 3.
6) Specification of query functionality for version 3.