Department of Veterans Affairs
Decentralized Hospital Computer Program

DHCP
HEALTH LEVEL SEVEN
(HL7)
DEVELOPER MANUAL

Version 1.6

October 1995

IRM Field Office
Albany, New York

Preface

The DHCP Health Level Seven (HL7) software package provides an interface that allows DHCP applications to exchange healthcare data with other applications using the HL7 protocol. This manual provides information for use by DHCP software developers to develop an interface between DHCP application packages and the DHCP HL7 package.

Table of Contents

Introduction 1
Overview 1
The DHCP HL7 Package 2
Lower Level Protocols 3
The DHCP Interface to the HL7 Protocol 4
Organization of this Manual 5
Related Manuals 6

Package Management 7

DHCP Interface Guide 9
General Overview 9
Part 1. Developing an Interface Specification 10
Part 2. Setting Up File Entries 14

Receiving HL7 Messages 23

Sending HL7 Messages 31

Reference Guide 37
Variables 37
Entry Points 43

Glossary 49

Appendix A. HL7 Interface Specification 51
Appendix B. Supported HL7 Message Types 57
Appendix C. Supported HL7 Segment Types 59
Appendix D. HL7/VA FileMan Data Types 63
Appendix E. HL7 Batch Protocol 65
Appendix F. Security 67

Index 69

Introduction

Overview

The first step in understanding the DHCP HL7 package is a basic understanding of HL7 itself. Health Level Seven (HL7) is a standard protocol which specifies the implementation of interfaces between two computer applications (sender and receiver) from different vendors for electronic data exchange in healthcare environments. HL7 allows healthcare institutions to exchange key sets of data from different application systems. Specifically, it defines the following:

> The data to be exchanged
> The timing of the interchange
> The communication of errors to the application.

The formats are generic in nature and must be configured to meet the needs of the two applications involved. An HL7 interface specification should be written detailing what formats (events, messages, segments, and fields) will be used, and the lower level protocol that will be implemented in order for the two applications to interface with one another. Appendix A of this manual is an example of an HL7 interface specification.

The HL7 protocol defines the content and format of abstract messages and transactions for interface capabilities for the following areas:

> Admission, discharge, and transfer (ADT)
> Order entry
> Query
> Financial applications such as charge, payment adjustments, and insurance
> Ancillary data reporting for Laboratory, Radiology, Pharmacy, etc.

In HL7, information is exchanged using HL7 messages when an event occurs in an application. Each HL7 message consists of one or more HL7 segments. A segment can be thought of as a record in a file. Each segment consists of one or more fields separated by a special character called the *field separator*. The field separator character is defined in the Header segment of an HL7 message. The Header segment is always the first segment in every HL7 message. Each field is assigned an HL7 data type (e.g., numeric, text, etc.).

Overview, cont.

In addition to the field separator character, there are four other special characters called *encoding characters*. Encoding characters are also defined in the Header segment. They operate on a single field in an HL7 segment. Each encoding character must be unique, and serves a specific purpose. None of the encoding characters can be the same as the field separator character.

> The first encoding character is the *component separator*. Some data fields can be divided into multiple components. The component separator is used to separate adjacent components within a data field.

> The second encoding character is the *repetition separator*. Some data fields can be repeated multiple times in a segment. The repetition separator character is used to separate multiple occurrences of a field.

> The third encoding character is the *escape character*. Data fields defined as text or formatted text can include escape sequences. The escape character is used to separate escape sequences from the actual text.

> The fourth encoding character is the *sub-component separator*. Some data fields can be divided into components, and each component can be further divided into sub-components. The sub-component separator is used to separate adjacent sub-components within a component of a field.

The DHCP HL7 Package

The purpose of the DHCP HL7 package is to assist DHCP applications in exchanging healthcare information with other applications using the HL7 protocol. The DHCP HL7 package consists of a set of utility routines and files that provide a generic interface to the HL7 protocol for all DHCP applications. The DHCP HL7 package can be divided into two parts:

> Lower level protocol support between sending and receiving applications
> DHCP interface to the HL7 protocol

Lower Level Protocols

The term lower level refers to a portion of the Open Systems Interconnect (OSI) model. The OSI model is divided into seven layers or levels. The lower levels (layers 1 through 4) support the actual movement of data between systems. This includes the actual physical connection between the systems and the communications protocol used.

The DHCP HL7 package supports the following lower level interfaces:

> HL7 Hybrid Lower Layer Protocol over an RS-232 connection
> DHCP MailMan messages
> X3.28 protocol

Using these lower level interfaces, the DHCP HL7 package can support layers 1 through 4 of the OSI model and eliminate the need for DHCP applications to write lower level interfaces each time they want to exchange data with another application.

These lower level interfaces provide the following functions:

> Receive and send HL7 messages
> Validate the HL7 message header information
> Invoke the appropriate DHCP application routine to process the data in the message
> Send HL7 accept acknowledgment (ACK) messages back to the sending application

The DHCP Interface to the HL7 Protocol

With the release of V. 1.6, DHCP HL7 supports several methods for interfacing to the HL7 protocol. The method established by V. 1.5 is still supported (for backwards compatibility), and a new method is introduced, as well as new routines, file structures, templates, menus, and options. There are some significant differences between the V. 1.5 and V. 1.6 interface methods, as shown in the following table.

V. 1.5 Interface Method	V. 1.6 Interface Method
One sender and one receiver per message.	One sender, one or more receivers.
Sender and receiver must be on different systems.	Sender and receiver can be on the same or different systems.
Messages must go through a communications protocol.	Messages sent to applications on the same system do not have to go through a communications protocol.
All messages are processed in the background.	Messages are processed in either the foreground or background, based on the priority assigned by sending/receiving applications.
No support for event points.	Event points are supported.

The DHCP HL7 package assists DHCP applications in interfacing to the HL7 protocol. In addition to the lower levels mentioned previously, all applications must perform the following upper level functions in order to exchange data with another application:

> Event analysis
> Data extraction
> Data filing
> Data formatting
> Message administration

Currently, the functions of event analysis, data extraction, and data filing must be performed by each application package. The DHCP HL7 package provides the following utilities to assist the application package with data formatting:

> Creation of HL7 Message Header (MSH and BHS) segments
> Utility calls to convert HL7 data to VA FileMan formats and vice versa
> Validation of message header information for all HL7 messages received
> A set of pre-defined variables for use in building HL7 messages/segments

The DHCP Interface to the HL7 Protocol, cont.

The DHCP HL7 package provides the following functions to assist the application package with message administration:

> Support for tracking transmissions and providing a status for each
> Generation of reports on pending transmissions and transmissions with errors
> A queue for incoming and outgoing transmissions
> A real-time monitor to monitor active transmission links and their statuses

Organization of this Manual

This manual is divided into the following sections:

> *Preface* - States the purpose of the software and the intended audience for this manual.
> *Table of Contents* - Lists the topics in the order in which they appear in this manual.
> *Introduction* - Provides an overview of the software and its purpose, refers you to related manuals, and explains the organization of this manual.
> *DHCP Interface Guide* - Provides step-by-step instructions for developing interfaces and setting up file entries.
> *Receiving HL7 Messages* - Provides instructions for receiving and processing HL7 messages.
> *Sending HL7 Messages* - Provides instructions for building HL7 messages to be transmitted to a receiving application.
> *Reference Guide* - Describes package variables and supported entry points.
> *Glossary* - Provides a list of terms used in this manual and their definitions.
> *Appendices* - Contain supplemental information.
> *Index* - Provides an alphabetical listing of the topics presented in this manual.

Related Manuals

For applications using the V. 1.6 interface method, you might also want to refer to the following manuals:

> DHCP HL7 V. 1.6 Installation Guide
> DHCP HL7 V. 1.6 Package Security Guide
> DHCP HL7 V. 1.6 Release Notes
> DHCP HL7 V. 1.6 Technical Manual
> DHCP HL7 V. 1.6 User Manual

For applications using the V. 1.5 interface method, you might also want to refer to the following manuals:

> DHCP HL7 V. 1.5 Developer Manual
> DHCP HL7 V. 1.5 Installation Guide
> DHCP HL7 V. 1.5 Package Security Guide
> DHCP HL7 V. 1.5 Release Notes
> DHCP HL7 V. 1.5 Technical Manual
> DHCP HL7 V. 1.5 User Manual

Package Management

1. The DHCP HL7 package consists of a set of utility routines and files that provide a generic interface to the HL7 Protocol for DHCP packages. HL7 allows healthcare institutions to exchange key sets of data from different application systems. Proper handling of this information is important to ensure patient confidentiality.

2. Per VHA Directive 10-93-142 regarding security of software, do not modify the DHCP HL7 routines and data dictionaries due to the high level of package integration.

DHCP Interface Guide

General Overview

This Interface Guide is divided into two parts:

> Part 1. Developing an Interface Specification
Provides step-by-step instructions for creating an API that allows your application to exchange information with other applications using the HL7 protocol.

> Part 2. Setting up file entries in the DHCP HL7 package
Provides step-by-step instructions for creating the appropriate file entries for the server applications that will originate messages and their clients (subscribers).

A DHCP application can interface with the DHCP HL7 package routines in two ways:

> To receive and process incoming HL7 messages. (Please refer to the Receiving Messages section of this manual.)
> To transmit outgoing HL7 messages. (Please refer to the Sending Messages section of this manual.)

Part 1. Developing an Interface Specification

Overview

The development of an interface specification details what data is available from another application and how to communicate to another application. The following steps will guide you in generating this important document. (Please refer to Appendix A of this manual to see a sample interface specification.)

Step	Description

1. Determine the events that trigger the exchange of HL7 messages with other applications. Generally, events like the following will trigger the exchange of HL7 messages in the form of reports or results:

   > Adding data in DHCP files
   > Modifying data in DHCP files
   > Canceling data in DHCP files
   > Releasing data in DHCP files

2. Analyze the data that your application will be exchanging with other applications. (Look at the fields in your application files and determine which DHCP fields would be of interest to other applications [or your same application at another facility]).

3. Determine whether there are corresponding fields and segments in the HL7 protocol. (To do this, look at the Health Industry Level 7 Interface Standards V. 2.2 Manual.)

4. Compare the DHCP fields with their corresponding HL7 fields and segments. Note any DHCP fields that need to be converted in order to correspond to HL7 field definitions and vice versa. (Please refer to Appendix C of this manual for a list of the HL7 segment types supported by this version of the DHCP HL7 package.)

Part 1. Developing an Interface Specification, cont.

Step Description

5. For DHCP fields that do not have corresponding HL7 fields, create locally defined HL7 fields and definitions using the HL7 data types (locally defined data types are not allowed). (Please refer to Appendix D of this manual for a list of HL7 data types and their corresponding VA FileMan data types.)

   The DHCP HL7 package supports all HL7 fields defined in V. 2.2 of the HL7 protocol, as well as locally defined fields used in locally defined segments. DHCP applications are not required to enter HL7 fields they will be using into the FIELD file (#771.1). This file is currently used only for documentation purposes.

   NOTE: Locally created fields and segments should be avoided wherever possible. Fields that are assigned an HL7 data type of ID must be assigned a locally created table number. Locally created table numbers must be six characters in length and begin with the 2-4 character DHCP package namespace followed by a sequential number (e.g., RA0001).

6. Logically group these locally created HL7 fields together and create local "Z"-type HL7 segments for them. Each segment must be given a 3-character name that starts with a "Z". Locally created segments and fields should be coordinated with the Albany IRMFO to avoid duplication. Locally defined segments are supported if they are entered into the SEGMENT NAME file (#771.3). It is recommended that packages enter locally defined HL7 segments into File #771.3 through use of the VA FileMan ^DIE utility in the application package post-init.

   NOTE: Some HL7 segments (such as those segments whose data elements in DHCP are the responsibility of a single DHCP application [e.g., MAS data]) can be usable by multiple applications. Once the responsible DHCP application has entered the segment in File #771.3 (along with a processing routine), other DHCP applications can use the segment processing routine by establishing an integration agreement with the responsible DHCP application.

7. Determine the HL7 messages that will be used to exchange data with other applications, and where the locally created "Z" type segments will be added to the HL7 segments that are already included in the HL7 messages. The HL7 message types supported by this version of the DHCP HL7 package are listed in Appendix B. Locally defined message types are not allowed.

Part 1. Developing an Interface Specification, cont.

Step	Description

8. Write an interface specification that documents all of this information. (Please refer to Appendix A of this manual to see a sample interface specification.) The interface specification you prepare will be given to other applications that will be exchanging information with your application. It will serve as the basic specification for interfacing with your application using the HL7 protocol.

   NOTE: If your HL7 interface specification specifies the use of the HL7 Hybrid Lower Layer Protocol, you will need to establish a port-to-port connection between the two application systems.

9. Write a segment driver routine for each HL7 segment (including locally defined "Z"-type segments) that you included in your interface specification. It is recommended that these routines be written as sub-routines with parameter passing and be modular in design. Each routine should be written to handle the processing of all fields in a specific segment. For outgoing HL7 messages, the routine should allow for a parameter to be passed that specifies the fields that should be processed and returned in the segment that the routine is building. For incoming HL7 messages, a parameter should be passed that is equal to the segment (which includes field values) that should be analyzed. The processing routine would convert the fields in the segment to DHCP format (and possibly file the fields in a DHCP file).

10. Write a message driver routine for each HL7 message that you included in your interface specification. The message driver routines for incoming and outgoing messages will be invoked by the DHCP HL7 package. Each application should create trigger event points in their applications that invoke an event driver protocol (in the PROTOCOL file [#101]) at appropriate locations where trigger events occur.

    The purpose of the message driver routines for outgoing HL7 messages is to

    > Determine the segments that are to be included in the HL7 message.
    > Determine the fields that are to be included in the HL7 segments.
    > Call the segment driver routine passing the appropriate variables.

Part 1. Developing an Interface Specification, cont.

Step	Description
10. cont.	Message driver routines for incoming HL7 messages operate in much the same manner as message driver routines for outgoing HL7 messages, except that the incoming message driver routine passes the HL7 segment (which includes field values) to be analyzed by the segment driver routine instead of a list of fields to be included in the segment.

Part 2. Setting up File Entries

Overview

Each application creates an entry in the HL7 APPLICATION PARAMETER file (#771), then every event related to the application is defined in the PROTOCOL file (#101) as a server. Other applications that require information related to the events become subscribers (clients) to those servers. They also have entries in File #771 and in File #101.

Scenario - Setting up File Entries for PIMS in DHCP HL7

This scenario is intended to assist you in following the step-by-step instructions beginning on the next page. The list numbers in the scenario below correspond with the step numbers in the instructions on the following pages.

1. An entry is made in File #771 defining the application parameters. For each application that wants to subscribe to the PIMS admission, discharge, and transfer events (ADT), an entry is made in the HL7 APPLICATION PARAMETER file (#771).
2. The parameters associated with the lower layer protocol (LLP) used by the client are entered in the HL LOGICAL LINK file (#870) and are associated with the subscribers.
3. The events for admission, discharge, and transfer (ADT) are defined in the PROTOCOL file (#101) as server protocols associated with the PIMS application. In addition, subscriber protocols are defined in the PROTOCOL file (#101). Lastly, the subscribers are added to the item multiple of the event protocol.

When a message is generated by a PIMS event, all the subscribers contained in the item multiple of the server protocol receive the message. If a DHCP application receives a message, the processing routine associated with the application is invoked. All acknowledgments are received and processed by the server protocol associated with the application.

Part 2. Setting up File Entries, cont.

The following steps describe what file entries must be set up prior to establishing communications. To set up an application, use the Interface Workbench option on the V1.6 OPTIONS menu. (For more information about setting up an application, please refer to the Interface Workbench option in the DHCP HL7 V. 1.6 User manual.)

Step	Description
1	Create an entry in the APPLICATION PARAMETER file (#771) for the server application that will originate the message. Additionally, each subscriber application will need to have an entry in this file. The entry you make in File #771 will need to be distributed by the application package to each site and entered in the HL7 APPLICATION PARAMETER file (#771) through the package post-init. Use the Interface Workbench option to edit the following fields in the APPLICATION PARAMETER file (#771):

NAME	This entry must be unique. For DHCP applications, it should be namespaced for your package.
ACTIVE/INACTIVE	For test purposes, it is recommended that this flag be set to *inactive*, and changed to *active* during the post-init or by user action subsequent to the post-init.
FACILITY NAME	This name can be any name desired for applications external to DHCP. For DHCP applications, the facility station number should be entered in this field.
COUNTRY CODE	This field is optional. Currently only the USA country code can be entered.
HL7 FIELD SEPARATOR	This field is optional. The HL7 field separator character to be used in building HL7 segments to send, or extracting fields of data from HL7 messages that have been received. The field separator is only one character (e.g., ^).

Part 2. Setting up File Entries, cont.

Step	Description	
1. cont.	HL7 ENCODING CHARACTERS	This field is optional. The HL7 encoding characters (1 to 4 characters) to be used in extracting data from HL7 segments and fields. Each character must be unique and cannot match the HL7 field separator character (described in the preceding paragraph). The four encoding characters are the component separator, repetition separator, escape character, and sub-component separator, in that order. The default characters used by the DHCP HL7 package (when an application package does not define its own encoding characters) are ~|\&.

2. Create an entry for each subscriber application in the HL LOGICAL LINK file (#870) if they are using a communication protocol to exchange data with the server application. If no communication protocol is used (i.e., two DHCP applications exchanging data on the same system), no entry is made in File #870. Use the Interface Workbench option to create an entry in File #870. Depending on the LLP type selected, you will be prompted for the following information:

   The parameters prompted for will depend on the Lower Layer Protocol (LLP) selected. The following are all the fields for the supported LLP in HL7 V. 1.6.

NODE	The name of the Logical Link you are creating.
QUEUE SIZE	The maximum number of transmissions to retain in the queue for historical purposes.
LLP TYPE	Currently supported LLP types are HLLP, MailMan, and X3.28.

Mail Man LLP

MAILGROUP	Mail Group to deliver messages.

Part 2. Setting up File Entries, cont.

Step	Description	
2. cont.	HLLP	
	DEVICE	Name of device for this logical link.
	VERSION ID	Version number used by the HLLP.
	BLOCK SIZE	Number of characters for a block of data.
	READ TIME-OUT	Number of seconds the Lower Layer Protocol remains in a state for data to come into the port.
	ACK TIME-OUT	The number of seconds the Lower Layer Protocol remains in a state for a low level acknowledgment.
	RE-TRANSMISSION ATTEMPTS	Number of time to retry sending the message.
	START BLOCK CHARACTER	This is the control character defined by the HLLP specification as the START BLOCK character.
	END BLOCK CHARACTER	This is the control character defined by the HLLP specification as the END BLOCK character.
	X3.28 -	
	DEVICE	Device for this Logical Link.
	BLOCK SIZE	Maximum Block size for a message.
	MESSAGE SIZE	Maximum message size.
	RESPONSE TIMER	Time in seconds for the response timer.
	RECEIVE TIMER	Time in seconds for the receive timer.
	INTER-BLOCK TIMER	Time in seconds for the inter-block timer.
	LINE CHECK TIMER	Time in seconds for Line check timer.

Part 2. Setting up File Entries, cont.

Step	Description	
2. cont.	SO	This is the control character defined by the LLP specification as the Start of Header character.
	ST	This is the control character defined by the LLP specification as the Start of Text character.
	ETB	This is the control character defined by the LLP specification as the End of Block character.
	ETX	This is the control character defined by the LLP specification as the End of Text character.
	EOT	This is the control character defined by the LLP specification as the End of Transmission character.
	ENQ	This is the control character defined by the LLP specification as the Enquiry character.
	TERM	This is the control character defined by the LLP specification as the Termination character.
	NAK	This is the control character defined by the LLP specification as the Negative Acknowledgment character.
	ACK0	This is the control character defined by the LLP specification as the Acknowledgment for block zero.
	ACK1	This is the control character defined by the LLP specification as the Acknowledgment for block one.
	ACK2	This is the control character defined by the LLP specification as the Acknowledgment for block two.
	ACK3	This is the control character defined by the LLP specification as the Acknowledgment for block three.
	ACK4	This is the control character defined by the LLP specification as the Acknowledgment for block four.

Part 2. Setting up File Entries, cont.

Step	Description	
2. cont.	ACK5	This is the control character defined by the LLP specification as the Acknowledgment for block five.
	ACK6	This is the control character defined by the LLP specification as the Acknowledgment for block six.
	ACK7	This is the control character defined by the LLP specification as the Acknowledgment for block seven.

3. Use the Interface Workbench option to create entries in the PROTOCOL file (#101) for both the server application that will generate HL7 messages and the subscriber (client) application that will receive the messages. You will be prompted for the following fields:

NAME	The name of the protocol. Must be namespaced for the application.
ITEM TEXT	A short description of the protocol.
DESCRIPTION	A full description of the protocol. It provides documentation to the local IRM staff and should be used to describe what this protocol is used for, the type of data exchanged, the reason for the exchange, etc.
PACKAGE	The DHCP package responsible for this protocol.
ENTRY ACTION	M code that will be executed before the Messaging System takes action on this protocol.
EXIT ACTION	M code that will be executed after the Messaging System is finished taking action on this protocol.
TYPE	The type of protocol (E for Event Driver, S for Subscriber).

Part 2. Setting up File Entries, cont.

Step	Description
3. cont.	Subsequent prompts depend on the type of protocol that was selected (Event Driver or Subscriber).

Fields prompted for *Event Driver* protocol	Fields prompted for *Subscriber* protocol
SERVER APPLICATION The application that will be sending the message. (Pointer to the APPLICATION PARAMETER file [#771].)	CLIENT (SUBSCRIBER) The application that will be receiving the message. (Pointer to the APPLICATION PARAMETER file [#771].)
MESSAGE TYPE The type of message that will be sent.(Pointer to the MESSAGE TYPE file [#771.2].)	MESSAGE TYPE The type of message that will be received. (Pointer to the MESSAGE TYPE file [#771.2].)
EVENT TYPE The type of event for which the message will be sent. (Pointer to the EVENT TYPE file [#779.001].)	EVENT TYPE The type of event for which the message will be received. (Pointer to the EVENT TYPE file [#779.001].)
PRIORITY The priority of the message for sending purposes: I = Immediate D = Delayed (Optional)	PRIORITY The priority of the message for receiving purposes: I = Immediate D = Delayed (Optional)
PROCESSING ID How the message should be processed: T = Training P = Production	LOGICAL LINK The transmission link over which the message will be sent/received. This field should not be filled in for two DHCP applications that will be exchanging messages on the same system. (Optional)

Part 2. Setting up File Entries, cont.

Step	Description	
3. cont.	Fields prompted for Event Driver protocol	Fields prompted for Subscriber protocol
	ACCEPT ACK CODE The manner in which the message should be acknowledged by the system that receives it. (Pointer to to the ACCEPT/APPLICATION ACK CONDITION file [#779.003].)	VERSION ID The version of the protocol standard that is being used (e.g., 2.2).
	APPLICATION ACK TYPE The manner in which the message should be acknowledged by the application that receives it. (Pointer to the ACCEPT/APPLICATION ACK CONDITION file [#779.003].)	SENDING FACILITY REQUIRED? Indicates whether the SENDING FACILITY field of the message header segment must be filled in. (Optional)
	VERSION ID The version of the protocol standard that is being used (e.g., 2.2).	RECEIVING FACILITY REQUIRED? Indicates whether the RECEIVING FACILITY field of the message header segment must be filled in. (Optional)
	PROCESS ACKNOWLEDGMENT ROUTINE The routine that will be invoked by the Messaging System to process an acknowledgment message if one is received. (Optional)	SECURITY REQUIRED? Indicates whether the SECURITY field of the message header segment must be filled in. (Optional)
	ITEM Subscriber protocols to this event driver protocol. (Can be one or more.) (Item multiple)	DATE/TIME OF MESSAGE REQUIRED? Indicates whether the DATE/TIME OF MESSAGE field of the message header segment must be filled in. (Optional) PROCESS ROUTINE - The routine that will be invoked by the Messaging System to process a message when it is received. This field must be filled in.

Receiving HL7 Messages

Overview

To receive and process incoming HL7 messages, the DHCP application must develop a M routine (or, optionally, an entry point in a routine) for each type of HL7 message it will receive. The name of the M routine (or, optionally, an entry point in a routine) is entered into the PROCESSING ROUTINE field of the PROTOCOL file (#101) of the application's subscriber protocol. The M routine should perform the following steps in order to receive and process an HL7 message:

Step	Description

1. When an HL7 message is received by the DHCP HL7 package, the ENTRY ACTION field of the PROTOCOL file (#101) is executed, then the Message Header segment (MSH) or Batch Header segment (BHS) is checked for validity. If an error is found, an HL7 acknowledgment message is sent to the sending application to report the error. If the header segment passes the validity checks, the following variables are set, and the DHCP application processing routine is invoked:

HL("APAT")	The application acknowledgment condition from the message header of the message received. This variable will be used by the receiving application to determine the type of acknowledgment, if any, that must be returned to the application that sent the message. (Optional)
HL("CC")	The country code from the message header of the message received. (Optional)
HL("DTM")	The date/time from the message header of the message received in HL7 format. (Optional)
HL("DUZ")	If a valid DHCP access code is contained in the first component of the SECURITY field (#8) of the MSH segment, HLDUZ will equal the DUZ associated with this access code from the NEW PERSON file (#200) on DHCP. (Optional)

Step	Description	
1. cont.	HL("ECH")	The HL7 encoding characters (one to four characters) to be used in extracting data from HL7 segments and fields. Each character must be unique and cannot match the HL7 field separator character. (See the variable HLFS for a definition of the field separator character.) The four encoding characters are the component separator, repetition separator, escape character, and sub-component separator, in that order. The default characters used by the DHCP HL7 package (when an application package does not define its own encoding characters) are ~|\&.
	HL("EID")	The IEN of the event driver protocol from the PROTOCOL file (#101) that generated the message
	HL("EIDS")	The IEN of the subscriber protocol from the PROTOCOL file (#101) that is receiving the message.
	HL("ESIG")	This variable might not always exist. If a valid DHCP electronic signature code is contained in the third component of the SECURITY field (#8) of the MSH segment, HLESIG will equal the signature block printed name associated with this electronic signature code from the NEW PERSON file (#200) on DHCP.
	HL("ETN")	The three character event type for the message received (e.g., A01 [Admit a Patient], O01 [Order Message], etc.).
	HL("FS")	The HL7 field separator character to be used in extracting fields of data from HL7 messages that have been received. The field separator is only one character (e.g., ^).
	HL("MID")	The HL7 message control ID for the message received. A number that uniquely identifies the message.

Step	Description	
1. cont.	HL("MTN")	The three character message type for the message received (e.g., QRY [Query], ORU [Observation Result Unsolicited], etc.).
	HL("PID")	The HL7 processing ID for the message received. (Normally, P for production, T for Training, D for debug.)
	HL("Q")	Two quotation marks (""). This variable can be used to insert a null value in an HL7 field when building HL7 segments.
	HL("RAN")	The name of the receiving application from the HL7 APPLICATION PARAMETER file (#771)(e.g., Radiology).
	HL("SAN")	The name of the sending application for the message received from the HL7 APPLICATION PARAMETER file (#771).
	HL("VER")	The version number of the HL7 protocol that was used to build the message being received.

2. All messages received are stored in a multiple field of the MESSAGE TEXT file (#772). This is the data that the receiving DHCP application must process. To process this data, the application must use the following three special variables:

HLNEXT	M code that is executed by the application to $O through the nodes of the Message Text global.
HLNODE	A node from the Message Text global.
HLQUIT	A variable that indicates when there are no more nodes to process. If HLQUIT'>0 all message text has been processed.

Step Description

3. The DHCP application processing routine should analyze the data in the message stored in the Message Text global and determine the appropriate HL7 response. To analyze the data, the application processing routine should execute the HLNEXT variable, analyze the data stored in the HLNODE variable, and quit if the HLQUIT variable is not greater than zero.

   The following is an example of possible application code:

```
N I,J,X
F I=1:1 X HLNEXT Q:HLQUIT'>0  S X(I)=HLNODE,J=0  F  S J=$O(HLNODE(J))
Q:'J  S X(I,J)=HLNODE(J)
```

   NOTE: For segments greater than 245 characters, HLNODE(X) will be defined.

   To respond to the incoming message, the DHCP application processing routine should set nodes of the HLA("HLA",I) local array equal to the appropriate HL7 segments or alternately, if the response is a large message, set HL7 segments into the global array ^TMP("HLA",$J,I). At this point, the receiving application becomes a sending application.

   NOTE: If a batch of HL7 messages (more than one) is to be created, the routine should invoke the CREATE^HLTF(HLMID,MTIEN,HLDT,HLDT1) entry point to obtain a message ID for the message being sent and to create an entry in the Message Text file (#772). The routine should then create the Message Header (MSH) segments for each HL7 message in the batch. The function call MSH^HLFNC2(HL,MID,RESULT,SECURITY) can be used to build the MSH segments. The message ID for each MSH segment should be created by concatenating together the message ID returned by the call to CREATE^HLTF, a hyphen, and a sequential whole number starting with the number 1 (e.g., 12345-1). If only one HL7 message is being created, the routine should not make the call to CREATE^HLTF or create the MSH segment. The DHCP HL7 package will create the MSH segment for you. The application processing routine should then invoke the GENACK^HLMA1(HLEID,HLMTIENS,HLEIDS,HLARYTYP,HLFORMAT, HLRESLTA,HLMTIENA,HLP) entry point to send the acknowledgment message, then quit to pass control back to the DHCP HL7 package.

Step Description

3. cont. The input and output parameters for these entry points are as follows:

CREATE^HLTF(HLMID,MTIEN,HLDT,HLDT1)

Input Parameters

HLMID	The parameter in which the message ID will be returned. This parameter must be passed by reference.
MTIEN	The parameter in which the IEN of the entry in the MESSAGE TEXT file (#772) will be returned. This parameter must be passed by reference.
HLDT	The parameter in which the message date/time in VA FileMan format will be returned. This parameter must be passed by reference.
HLDT1	The parameter in which the message date/time in HL7 format will be returned. This parameter must be passed by reference.

Output Parameters

All of the above input parameters are returned as output parameters.

MSH^HLFNC2(HL,MID,RESULT,SECURITY)

Input Parameters

HL	The array in which the output parameters will be returned. This parameter must be passed by reference.
MID	The parameter in which the message ID will be returned.

Step Description

3. cont. RESULT — The message ID assigned to this message and/or an error message will be returned in this parameter. This parameter must be passed by reference. If the call to MSH^HLFNC2 is successful, this parameter will be returned equal to the message ID assigned to the message that was created. If the call was not successful, this parameter will be returned with the following three pieces of data: message ID (or 0 if no message ID was assigned)^error code^error message.

SECURITY — Security information (1 to 40 characters) that the DHCP application wants included in the SECURITY field (#8) of the HL7 MSH or BHS segment when sending a message. (Optional)

Output Parameters

All of the above input parameters are returned as output parameters.

GENACK^HLMA1(HLEID,HLMTIENS,HLEIDS,HLARYTYP,HLFORMAT, HLRESLTA,HLMTIENA,HLP)

Input Parameters

HLEID — The IEN of the event driver protocol in the PROTOCOL file (#101). It is passed to the processing routine in the variable HL("EID")

HLMTIENS — The IEN of the entry in the MESSAGE TEXT file (#772) for the subscriber application.

HLEIDS — The IEN of the subscriber protocol in the PROTOCOL file (#101). It is passed to the processing routine in the variable HL("EIDS")

HLARYTYP — This parameter specifies where the acknowledgment array is stored and whether it is a single message or batch acknowledgment. It must equal LM for Local/Single Message, LB for Local/Batch Message, GM for Global/Single Message or GB for Global/Batch Message

Step	Description	
3. cont.	HLFORMAT	This parameter specifies whether the HLA array is pre-formatted in HL7 format. At this time, it should always equal 1.
	HLRESLTA	The message ID assigned to this message and/or an error will be returned in this parameter. This parameter must be passed by reference.
	HLMTIENA	The IEN of the entry in the MESSAGE TEXT file (#772) created by the call to the entry point CREATE^HLTF and returned in the MTIEN parameter.
	HLP("ERRTEXT")	If an error occurred during the processing of the incoming message, an error message (1 to 80 characters) should be passed in this parameter. (Optional)
	HLP("PRIORITY")	The default priority is delayed. Set this parameter equal to "I" for Immediate if this message should be delivered immediately. (Optional)
	HLP("SECURITY")	Security information (1 to 40 characters) that the DHCP application wants included in the SECURITY field (#8) of the HL7 MSH or BHS segment when sending a message. (Optional)
	HLA("HLA",I)	A local array consisting of HL7 segments that form an HL7 message where the variable I is a sequential, whole number starting with the number one. This array is built by the DHCP application in order to send an HL7 message that is small enough to be built in the local partition space. Otherwise, the ^TMP("HLA") global array defined below should be set.
	^TMP("HLA",$J,I)	A global array containing all segments of the HL7 message that the receiving DHCP application wishes to send as a response. The variable I is a sequential, whole number starting with the number one.

Step	Description
3. cont.	GENACK^HLMA1(HLEID,HLMTIENS,HLEIDS,HLARYTYP,HLFORMAT, HLRESLTA,HLMTIENA,HLP)

Output Parameters

HLRESLTA	If the call to GENACK is successful, this parameter will be returned equal to the message ID assigned to the message that was created. If the call was not successful, this parameter will be returned with the following three pieces of data: message ID (or 0 if no message ID was assigned)^error code^error message.

Sending HL7 Messages

Overview

To transmit HL7 messages, the DHCP application must develop a M routine (or, optionally, an entry point in a routine) for each type of HL7 message it will be sending. (Please refer to Appendix B for a list of supported HL7 message types.) The M routine should perform the following steps in order to build an HL7 message for transmission to the receiving application:

Step	Description

1. Invoke the subroutine entry point INIT^HLFNC2(EID,HL,INT) to initialize variables needed to build a message. The input and output parameters for this call are as follows:

   Input Parameters

EID	The IEN of the event driver protocol in the PROTOCOL file (#101) for the application that is sending this message.
HL	The array in which the output parameters will be returned. This parameter must be passed by reference.
INT	Indicates that only array values for an internal DHCP-to-DHCP message exchange should be utilized.

   Output Parameters

HL("ACAT")	The accept acknowledgment type from the PROTOCOL file (#101). (Optional)
HL("APAT")	The application acknowledgment condition from the PROTOCOL file (#101).(Optional)
HL("CC")	The country code of the sending application from the HL7 APPLICATION PARAMETER file (#771). (Optional)

Step	Description	
1. cont.	HL("ECH")	The HL7 encoding characters (one to four characters) to be used in extracting data from HL7 segments and fields. Each character must be unique and cannot match the HL7 field separator character. (See the variable HLFS for a definition of the field separator character.) The four encoding characters are the component separator, repetition separator, escape character, and sub-component separator, in that order. The default characters used by the DHCP HL7 package (when an application package does not define its own encoding characters) are ~|\&.
	HL("ETN")	The event type name from the PROTOCOL file (#101) (e.g., A01).
	HL("FS")	The HL7 field separator character to be used in building HL7 segments. The field separator is only one character (e.g., ^).
	HL("MTN")	The message type name from the PROTOCOL file (#101) (e.g., ADT).
	HL("PID")	The HL7 processing ID for the message being sent. (Normally, P for production, T for Training, D for debug.)
	HL("Q")	Two quotation marks (""). This variable can be used to insert a null value in an HL7 field when building HL7 segments.
	HL("SAF")	The name of the sending facility from the DHCP APPLICATION PARAMETER file (#771).
	HL("SAN")	The name of the sending application from the DHCP APPLICATION PARAMETER file (#771)(e.g., Radiology).
	HL("VER")	The version number of the HL7 protocol that was used to build the message being sent.

Step Description

2. Check for $O(HL("")). If this check proves false, the call to INIT^HLFNC2 failed. The HL variable will contain the error message/reason for failure. The routine should quit. Otherwise, the routine should set the local array HLA("HLS",I) or, alternately, if the HL7 message is a large one, the global array ^TMP("HLS",$J,I) equal to the HL7 segments to be included in the HL7 message. If a batch of HL7 messages (more than one) is to be created, the routine should invoke the CREATE^HLTF(HLMID,MTIEN,HLDT,HLDT1) entry point to obtain a message ID for the message being sent and to create an entry in the MESSAGE TEXT file (#772). The routine should then create the Message Header (MSH) segments for each HL7 message in the batch. The function call MSH^HLFNC2 can be used to build the MSH segments. The message ID for each MSH segment should be created by concatenating together the message ID returned by the call to CREATE^HLTF, a hyphen, and a sequential, whole number starting with 1 (e.g., 12345-1). If only one HL7 message is being created, the routine should not make the call to CREATE^HLTF or create the MSH segment. The DHCP HL7 package will create the MSH segment for you. The input and output parameters for CREATE^HLTF(HLMID,MTIEN,HLDT,HLDT1)are as follows:

   Input Parameters

HLMID	The parameter in which the message ID will be returned. This parameter must be passed by reference.
MTIEN	The parameter in which the IEN of the entry in the MESSAGE TEXT file (#772) will be returned. This parameter must be passed by reference.
HLDT	The parameter in which the message date/time in VA FileMan format will be returned. This parameter must be passed by reference.
HLDT1	The parameter in which the message date/time in HL7 format will be returned. This parameter must be passed by reference.

   Output Parameters

   All of the above input parameters are returned as output parameters.

Step Description

2. cont. The input and output parameters for MSH^HLFNC2(HL,MID,RESULT, SECURITY) are as follows:

Input Parameters

HL	The array in which the output parameters will be returned. This parameter must be passed by reference.
MID	The parameter in which the message ID will be returned.
RESULT	The message ID assigned to this message and/or an error message will be returned in this parameter. This parameter must be passed by reference. If the call to MSH^HLFNC2 is successful, this parameter will be returned equal to the message ID assigned to the message that was created. If the call was not successful, this parameter will be returned with the following three pieces of data: message ID (or 0 if no message ID was assigned)^error code^error message.
SECURITY	Security information (1 to 40 characters) that the DHCP application wants included in the SECURITY field (#8) of the HL7 MSH or BHS segment when sending a message. (Optional)

Output Parameters

All of the above input parameters are returned as output parameters.

The variables that the DHCP application is responsible for setting are defined as follows:

HLA("HLS",I)	A local array consisting of HL7 segments that form an HL7 message where the variable I is a sequential, whole number starting with the number one. This array is built by the DHCP application in order to send an HL7 message that is small enough to be built in the local partition space. Otherwise, the ^TMP("HLS") global array defined below should be set.

Step	Description	
2. cont.	^TMP("HLS",$J,I)	A global array containing all segments of the HL7 message that the receiving DHCP application wishes to send as a response. The variable I is a sequential, whole number starting with the number one.

3. Once the HLA("HLS") local array or the ^TMP("HLS") global array has been created, the routine should call the subroutine entry point GENERATE^ HLMA(HLEID,HLARYTYP,HLFORMAT,HLRESLT,HLMTIEN,HLP). The input and output parameters for this call are as follows:

Input Parameters

HLEID	The IEN of the event driver protocol in the Protocol file (#101). It is passed to the processing routine in the variable HL("EID")
HLARYTYP	This parameter specifies where the acknowledgment array is stored and whether it is a single message or batch acknowledgment. It must equal LM for Local/Single Message, LB for Local/Batch Message, GM for Global/Single Message or GB for Global/Batch Message
HLFORMAT	This parameter specifies whether the HLA array is pre-formatted in HL7 format. At this time, it should always equal 1.
HLMTIEN	The IEN of the entry in the Message Text file (#772) created by the call to the entry point CREATE^HLTF.
HLRESLT	The message ID assigned to this message and/or an error message will be returned in this parameter. This parameter must be passed by reference.
HLP("CONTPTR")	The value that should go in the continuation pointer field of the Message Header segment for the message being sent.

Step Description

3. Input Parameters, cont.

cont.

HLP("PRIORITY")	This parameter is optional. The default priority is delayed. Set this parameter equal to I for Immediate if this message should be delivered in the foreground (immediately).
HLP("SECURITY")	Security information that the DHCP application wants included in the Security field (#8) of the HL7 MSH or BHS segment when sending a message. This is an optional variable.

Output Parameters

HLRESLT	If the call to GENERATE^HLMA is successful, this parameter will be returned equal to the message ID assigned to the message that was created. If the call was not successful, this parameter will be returned with the following three pieces of data: message ID (or 0 if no message ID was assigned)^error code^error message.

4. When the entry point GENERATE^HLMA is invoked, the data in the HLA("HLS") local array or ^TMP("HLS") global array is loaded into the MESSAGE TEXT file (#772) and the entry in the MESSAGE TEXT file (#772) is completed. The message is then delivered to the subscribers to the event driver protocol specified in the PROTOCOL file (#101).

Reference Guide

Variables

The following table provides a list of the basic variables, with their descriptions, that are used by the DHCP HL7 package for the V. 1.6 interface method. The variables are grouped into the following three categories:

> Variables created when an HL7 message is *received*
> Variables created when an HL7 message is being *sent*
> Variables created when HL7 messages are both *sent and received*

Variable Name	Description	When Created
EID	The IEN of the event driver protocol in the PROTOCOL file (#101) for the application that is sending this message.	Sent
HL	The array in which the output parameters will be returned. *This parameter must be passed by reference.*	Sent and Received
HL("ACAT")	The accept acknowledgment type from the PROTOCOL file (#101). (Optional)	Sent
HL("APAT")	The application acknowledgment condition of the sending application from the PROTOCOL file (#101). It is in the message header of the message received. This variable will be used by the receiving application to determine the type of acknowledgment, if any, that must be returned to the application that sent the message. (Optional)	Sent and Received
HL("CC")	The country code of the sending application from the HL7 APPLICATION PARAMETER file (#771). It is in the message header of the message received. (Optional)	Sent and Received
HL("DTM")	The date/time from the message header of the message received in HL7 format. (Optional)	Received
HL("DUZ")	If a valid DHCP access code is contained in the first component of the SECURITY field (#8) of the MSH segment, HLDUZ will equal the DUZ associated with this access code from the NEW PERSON file (#200) on DHCP. (Optional)	Received

Variables, cont.

Variable Name	Description	When Created
HL("ECH")	The HL7 encoding characters (1 to 4 characters) to be used in extracting data from HL7 segments and fields. Each character must be unique and cannot match the HL7 field separator character. (See the variable HLFS for a definition of the field separator character.) The four encoding characters are the component separator, repetition separator, escape character, and sub-component separator, in that order. The default characters used by the DHCP HL7 package (when an application package does not define its own encoding characters) are ~|\&.	Sent and Received
HL("EID")	The IEN of the event driver protocol from the PROTOCOL file (#101) that generated the message.	Received
HL("EIDS")	The IEN of the subscriber protocol from the PROTOCOL file (#101) that is receiving the message.	Received
HL("ESIG")	This variable might not always exist. If a valid DHCP electronic signature code is contained in the third component of the SECURITY field (#8) of the MSH segment, HLESIG will equal the signature block printed name associated with this electronic signature code from the NEW PERSON file (#200) on DHCP.	Received
HL("ETN")	The 3 character event type name from the PROTOCOL file (#101) (e.g., A01 [Admit a Patient], O01 [Order Message], etc.).	Sent and Received
HL("FS")	The HL7 field separator character to be used in extracting fields of data from HL7 messages received, or building HL7 segments in messages sent. The field separator is only one character (e.g., ^).	Sent and Received
HL("MID")	The HL7 message control ID for the message received. A number that uniquely identifies the message.	Received
HL("MTN")	The three character message type name from the PROTOCOL file (#101) (e.g., ADT, QRY [Query], ORU [Observation Result Unsolicited], etc.).	Sent and Received
HL("PID")	The HL7 processing ID for the message received. (Normally, P for production, T for Training, D for Debug.)	Sent and Received
HL("Q")	Two quotation marks (""). This variable can be used to insert a null value in an HL7 field when building HL7 segments.	Sent and Received

Variables, cont.

Variable Name	Description	When Created
HL("RAN")	The name of the receiving application from the HL7 APPLICATION PARAMETER file (#771) (e.g., Radiology).	Received
HL("SAF")	The name of the sending facility from the HL7 APPLICATION PARAMETER file (#771).	Sent
HL("SAN")	The name of the sending application (e.g., Radiology) from the HL7 APPLICATION PARAMETER file (#771) for the message received.	Sent and Received
HL("VER")	The version number of the HL7 protocol that was used to build the message being sent/received.	Sent and Received
HLA("HLA",I)	A local array consisting of HL7 segments that form an HL7 message where the variable I is a sequential, whole number starting with the number 1. This array is built by the DHCP application in order to send an HL7 message that is small enough to be built in the local partition space. Otherwise, the ^TMP("HLA") global array should be set.	Received
HLA("HLS",I)	A local array consisting of HL7 segments that form an HL7 message where the variable I is a sequential, whole number starting with the number 1. This array is built by the DHCP application in order to send an HL7 message that is small enough to be built in the local partition space. Otherwise, the ^TMP("HLS") global array defined below should be set.	Sent
HLARYTYP	This parameter specifies where the acknowledgment array is stored and whether it is a single message or batch acknowledgment. It must equal LM for Local/Single Message, LB for Local/Batch Message, GM for Global/Single Message or GB for Global/Batch Message.	Sent and Received
HLDT	The parameter in which the message date/time in internal VA FileMan format will be returned. *This parameter must be passed by reference.*	Sent and Received
HLDT1	The parameter in which the message date/time in HL7 format will be returned. *This parameter must be passed by reference.*	Sent and Received
HLEID	The IEN of the event driver protocol in the PROTOCOL file (#101). It is passed to the processing routine in the variable HL("EID").	Sent and Received
HLEIDS	The IEN of the subscriber protocol in the PROTOCOL file (#101). It is passed to the processing routine in the variable HL("EIDS").	Received

Variables, cont.

Variable Name	Description	When Created
HLFORMAT	This parameter specifies whether the HLA array is pre-formatted in HL7 format. At this time, it should always equal 1.	Sent and Received
HLMID	The parameter in which the message ID will be returned. *This parameter must be passed by reference.*	Sent and Received
HLMTIEN	The parameter in which the IEN of the entry in the MESSAGE TEXT file (#772) created by the call to the entry point CREATE^HLTF.	Sent
HLMTIENA	The IEN of the entry in the MESSAGE TEXT file (#772) created by the call to the entry point CREATE^HLTF and returned in the MTIEN parameter.	Received
HLMTIENS	The IEN of the entry in the MESSAGE TEXT file (#772) for the subscriber application.	Received
HLNEXT	M code that is executed by the application to $O through the nodes of the Message Text global.	Received
HLNODE	A node from the Message Text global.	Received
HLP("CONTPTR")	The value that should go in the CONTINUATION POINTER field of the Message Header segment for the message being sent.	Sent
HLP("ERRTEXT")	If an error occurred during the processing of the incoming message, an error message (1 to 80 characters) should be passed in this parameter. (Optional)	Received
HLP("PRIORITY")	The default priority is delayed. Set this parameter equal to I for Immediate if this message should be delivered in the foreground (immediate).	Sent and Received
HLP("SECURITY")	Security information (1 - 40 characters) that the DHCP application wants included in the SECURITY field (#8) of the HL7 MSH or BHS segment when sending a message. (Optional)	Sent and Received
HLQUIT	A variable that indicates when there are no more nodes to process. If HLQUIT is not greater than zero, all message text has been processed.	Received

Variables, cont.

Variable Name	Description	When Created
HLRESLT	The message ID assigned to this message and/or an error message will be returned in this parameter. This parameter must be passed by reference. If the call to GENERATE^HLMA is successful, this parameter will be returned equal to the message ID assigned to the message that was created. If the call was not successful, this parameter will be returned with the following three pieces of data: message ID (or 0 if no message ID was assigned)^error code^error message.	Sent
HLRESLTA	The message ID assigned to this message and/or an error will be returned in this parameter. This parameter must be passed by reference. If the call to GENACK is successful, this parameter will be returned equal to the message ID assigned to the message that was created. If the call was not successful, this parameter will be returned with the following three pieces of data: message ID (or 0 if no message ID was assigned)^error code^error message.	Received
INT	Indicates that only array values for an internal DHCP-to-DHCP message exchange should be utilized.	Sent
MID	The parameter in which the message ID will be returned.	Sent and Received
MTIEN	The parameter in which the IEN of the entry in the MESSAGE TEXT file (#772) (created by the call to the entry point CREATE^HLTF) will be returned. *This parameter must be passed by reference.*	Sent and Received
MTIENA	The IEN of the entry in the MESSAGE TEXT file (#772) created by the call to the entry point CREATE^HLTF and returned in the MTIEN parameter.	Received
PRIORITY	The default priority is delayed. Set this parameter equal to I for Immediate if this message should be delivered in the foreground (immediately). (Optional)	Sent and Received

Variables, cont.

Variable Name	Description	When Created
RESULT	The message ID assigned to this message and/or an error message will be returned in this parameter. T*his parameter must be passed by reference.* If the call to MSH^HLFNC2 is successful, this parameter will be returned equal to the message ID assigned to the message that was created. If the call was not successful, this parameter will be returned with the following three pieces of data: message ID (or 0 if no message ID was assigned)^error code^error message.	Sent and Received
SECURITY	Security information (1 to 40 charac ters) that the DHCP application wants included in the SECURITY field (#8) of the HL7 MSH or BHS segment when sending a message. (Optional)	Sent and Received
^TMP("HLA",$J,I)	A global array containing all segments of the HL7 message that the receiving DHCP application wishes to send as a response. The variable I is a sequential, whole number starting with the number 1.	Received
^TMP("HLS",$J,I)	A global array containing all segments of the HL7 message that the receiving DHCP application wishes to send as a response. The variable I is a sequential, whole number starting with the number 1.	Sent

Entry Points

The following is a List of supported entry points into routines belonging to the DHCP HL7 package. These entry points should be used by individual DHCP packages using the V. 1.6 interface method. For each entry point listed, the following information is provided:

> Entry point name and description
> Required/optional input parameters
> Output parameters, if applicable

CREATE^HLTF(HLMID,MTIEN,HLDT,HLDT1)

If a batch of HL7 messages (more than one) is to be created, the application processing routine should invoke this entry point to obtain a message ID for the message being sent, and to create an entry in the MESSAGE TEXT file (#772).

Required Input Parameters:	HLDT, HLDT1, HLMID, MTIEN (These parameters must be passed by reference.)
Output Parameters:	All of the above input variables are returned as output variables.

GENACK^HLMA1(HLEID,HLMTIENS,HLEIDS,HLARYTYP,HLFORMAT, HLRESLTA,HLMTIENA,HLP)

After the MSH segment is created, the application processing routine should invoke this entry point to send the acknowledgment message, then quit to pass control back to the DHCP HL7 package. If the call to GENACK is successful, the HLRESLTA parameter will be returned equal to the message ID assigned to the message that was created. If the call was not successful, the HLRESLTA parameter will be returned with the following three prices of data: message ID (or 0 if no message ID was assigned)^error code^error message.

Required Input Parameters:	HLEID, HLMTIENS, HLEIDS, HLARYTYP, HLFORMAT, HLRESLTA, HLMTIENA, HLP (HLRESLTA must be passed by reference.)
Optional Input Parameters:	HLMTIENA, HLP("PRIORITY"), HLP("SECURITY")
Output Parameters:	HLRESLTA

Entry Points, cont.

GENERATE^HLMA(HLEID,HLARYTYP,HLFORMAT,HLMTIEN,HLRESLT,HLP)

When this entry point is invoked, it loads the data in the HLA("HLS") local array or the ^TMP("HLS") global array in the MESSAGE TEXT file (#772), and the entry in the MESSAGE TEXT file (#772) is completed. The message is then delivered to the subscribers to the event driver protocol specified in the PROTOCOL file (#101). If the call to GENERATE^HLMA is successful, the HLRESLT parameter will be returned equal to the message ID assigned to the message that was created. If the call was not successful, the HLRESLT parameter will be returned with the following three prices of data: message ID (or 0 if no message ID was assigned)^error code^error message.

Required Input Parameters:	HLEID, HLARAYTYP, HLFORMAT, HLRESLT (HLRESLT must be passed by reference.)
Optional Input Parameters:	HLMTIEN, HLP("PRIORITY"), HLP("SECURITY"), HLP("CONTPTR")
Output Parameters:	HLRESLT

INIT^HLFNC2(EID,HL,INT)

To transmit HL7 messages, the DHCP application must develop a MUMPS routine (or, optionally, an entry point in a routine) for each type of HL7 message it will be sending. (Please refer to Appendix B for a list of supported HL7 message types.) The MUMPS routine should invoke this subroutine entry point to initialize variables needed to build an HL7 message for transmission to the receiving application.

Input Parameters:	EID, HL (HL must be passed by reference.)
Optional Input Parameters	INT
Output Parameters:	HL("ACAT"), HL("APAT"), HL("CC"), HL("ECH"), HL("ETN"), HL("FS"), HL("MTN")HL("PID"), HL("Q"), HL("SAN"), HL("SAF"), HL("VER")

Entry Points, cont.

MSH^HLFNC2(HL,MID,RESULT,SECURITY

This is a function call used to build MSH segments if a batch of HL7 messages (more than one) is being created. The message ID for each MSH segment should be created by concatenating together:

1. The message ID returned by the call to CREATE^HLTF
2. A hyphen
3. A sequential, whole number starting with 1 (e.g., 12345-1).

If only one HL7 message is being created, the routine should not make the call to CREATE^HLTF or create the MSH segment. The DHCP HL7 package will create the MSH segment for you.

Required Input Parameters: HL, MID, RESULT

Optional Input Parameters: SECURITY

Output Parameter: RESULT

FMDATE^HLFNC(X)

This entry point is an extrinsic function call. It converts a date, date/time, or time in HL7 format to a date, date/time, or time in internal VA FileMan format. The variable X is the input variable equal to the HL7 date/time.

Required Input Variables: X

FMNAME^HLFNC(X)

This entry point is an extrinsic function call. It converts a name in HL7 format to a name in DHCP format. The variable X is the input variable equal to the HL7 name.

Required Input Variables: X

NOTE: The variable HLECH (HL7 encoding characters) must be defined before calling this entry point.

Entry Points, cont.

HLADDR^HLFNC(AD,GL)

This entry point is an extrinsic function call. It converts address information in DHCP format to an address in HL7 format. The input variable AD can be one to four components representing lines of a street address. The components must be separated by up-arrows (e.g., VA Medical Center^123 Anywhere Street^Suite 101^Building 36). The input variable GL can be one to four components contain geographic location information in the following order: city, state or province, zip code, country code. The components must be separated by up-arrows (e.g., city^state or province^zip code^country code). The maximum length of the formatted string of six address components that is returned (including component separators) will be 106 characters.

The first component of the GL variable (city) will be truncated to no more than 30 characters. The second component of the GL variable (state or province) must be the internal entry number of an entry in the STATE file (#5). The third component of the GL variable (zip code) must pattern match 5N, 9N or 5N1"-"4N. The fourth component of the GL variable (country code) will be set to "USA" if it is not defined.

NOTE: The variable HLECH (HL7 encoding characters) must be defined before calling this entry point.

Required Input Variables: AD and GL

HLDATE^HLFNC(X,Y)

This entry point is an extrinsic function call. It converts a date, date/time, or time in internal VA FileMan format to a date, date/time, or time in HL7 format. The variable X is the input variable equal to the VA FileMan date/time. The format that is returned (date only, date/time or time only) is normally dependent on the value of the input variable X. However, the format to be returned can be forced to a specific value by setting the optional variable Y equal to one of the following values:

Y = DT Date Only
Y = TM Time Only
Y = TS Date and Time

Required Input Variables: X

Optional Input Variables: Y

Entry Points, cont.

HLNAME^HLFNC(X)

This entry point is an extrinsic function call. It converts a name in DHCP format (e.g., lastname,firstname) to a name in HL7 format. The variable X is the input variable equal to the DHCP name.

Required Input Variables: X

NOTE: The variable HLECH (HL7 encoding characters) must be defined before calling this entry point.

HLPHONE^HLFNC(X,B,C)

This entry point is an extrinsic function call. It converts a phone number in DHCP format to a phone number in HL7 format. The input variable X is the DHCP phone number to be converted. Optional input variables are the variable B, equal to a beeper number, and the variable C, equal to a comment. The maximum length of the formatted string that is returned is 40 characters.

Required Input Variables: X

Optional Input variables: B and C

M10^HLFNC(X)

This entry point is an extrinsic function call. It calculates a checksum using the M10 algorithm for a value in the variable X and passes back the result in HL7 format (i.e., value~checksum~algorithm) (e.g., 55555~5~M10). The value of the input variable X should be the first component (the ID) of an HL7 field whose data type is COMPOSITE ID WITH CHECK DIGIT (CK).

Required Input Variables: X

NOTE: The variable HLECH (HL7 encoding characters) must be defined before calling this entry point.

Entry Points, cont.

M11^HLFNC(X)

This entry point is an extrinsic function call. It performs the same function as the call to M10^HLFNC, except it uses the M11 algorithm to calculate the checksum. The value of the input variable X should be the first component (the ID) of an HL7 field whose data type is COMPOSITE ID WITH CHECK DIGIT (CK).

Required Input Variables: X

Note: The variable HLECH (HL7 encoding characters) must be defined before calling this entry point.

UPPER^HLFNC(X)

This entry point converts lowercase characters in the input variable X to uppercase characters.

Required Input Variables: X

Glossary

DHCP Application	A software package developed by the VA to support clinical or administrative functions at VA medical centers nationwide. It is written in M and, via Kernel, will run on all major M implementations, regardless of vendor.
HL7 Component	A field can contain multiple components separated by the HL7 component separator.
HL7 Field	A field is a specific unit of data. Each field is defined by the following set of characteristics: Position in the Segment, Name, ID Number, Maximum Length, Optionality, Repetition, Table Assignment (optional) and Type.
HL7 Hybrid Lower Layer Protocol	A communication protocol that supports layers one through four of the OSI protocol.
HL7 Interface	The exchange of information between a DHCP application and the DHCP HL7 package.
HL7 Message	A message is the atomic unit for transferring data between systems. It is comprised of a group of HL7 segments in a defined sequence. Each message has a message type that defines its purpose. Each message is identified by a unique three character code.
HL7 Protocol	Health Level Seven. An application communications standard for text-type patient-specific data. Permits data exchange between diverse computer configurations with a variety of communications protocols. Communications take place by exchange of HL7 "messages".

HL7 Segment	A segment is a logical grouping of one or more data fields separated by the HL7 field separator. Segments of a message might be optional or required. They might occur only once or might repeat multiple times. Each segment is identified by a unique three character code.
Lower Level Interface	Refers to layers one through four of the Open Systems Interconnect (OSI) protocol for exchanging data between computer systems. Layers one through four ensure physical connectivity and error-free delivery of data between computer systems and are normally handled by a communication protocol independent of the HL7 protocol. In the DHCP HL7 package the lower level interface is handled by either the DHCP MailMan package or the HL7 Hybrid Lower Layer Protocol.
Non-DHCP Application	A term used to refer to and distinguish between the two applications (the other is called the DHCP application) that will be exchanging data using the HL7 protocol.

Appendix A. Sample HL7 Interface Specification

HEALTH LEVEL 7
INTERFACE SPECIFICATIONS
ALBANY INFORMATION SYSTEMS CENTER
DEPARTMENT OF VETERANS AFFAIRS

DECENTRALIZED HOSPITAL COMPUTER PROGRAM
EXCHANGE OF RADIOLOGY HEALTHCARE INFORMATION

MARCH 1993

1. PURPOSE

This document specifies an interface to the DHCP Radiology package based upon the HL7 protocol. It is intended that this interface form the basis for the exchange of healthcare information between the DHCP Radiology package and all non-DHCP systems, especially those non-DHCP systems that generate radiology results information.

2. OVERVIEW

2.1 Statement of Intent

The Albany IRM Field Office (IRMFO) is developing and plans to implement a generic interface to the HL7 protocol for use by the DHCP Radiology package in communicating with non-DHCP systems for the purpose of exchanging healthcare information. This interface might eventually be used by all DHCP clinical packages to exchange healthcare information with non-DHCP systems. The interface will strictly adhere to the HL7 protocol and will avoid using "Z" type extensions to the protocol wherever possible.

2.2 Scope

This document describes messages that are exchanged between the DHCP Radiology package and a non-DHCP system for the purpose of exchanging information concerning radiology results, specifically reports and impressions.

3. GENERAL SPECIFICATIONS

3.1 Communication Protocol

The HL7 protocol defines only the seventh level of the Open System Interconnect (OSI) protocol. This is the application level. Levels one through six involve primarily communication protocols. The HL7 protocol provides some guidance in this area. The communication protocols that will be used for interfacing with the DHCP Radiology package will be based on the HL7 Hybrid Lower Level Protocol which is described in the HL7 Interface Standards document.

3.2 Application Processing Rules

The HL7 protocol itself describes the basic rules for application processing by the sending and receiving systems. Information contained in the protocol will not be repeated here, therefore anyone wishing to interface with the DHCP Radiology package should become familiar with the HL7 protocol V. 2.1.

3.3 Messages

The following HL7 messages will be used to support the exchange of Radiology data:

ACK	General Acknowledgment
ORF	Observational Report Response
ORM	Order
ORR	Order Response Message
ORU	Observational Results Unsolicited
QRY	Query Message

3.4 Segments

The following HL7 segments will be used to support the exchange of Radiology data:

MSA	Message Acknowledgment
MSH	Message Header
OBR	Observational Request
OBX	Result
ORC	Common Order
PID	Patient Identification
QRD	Query Definition

3.5 Fields

The following HL7 fields will be used to support the exchange of Radiology data for each of the segments listed in paragraph 3.4:

SEGMENT	FIELD SEQUENCE NUMBER	FIELD ELEMENT NAME
MSA	1	Acknowledgment Code
	2	Message Control ID
	3	Text Message
MSH	1	Field Separator
	2	Encoding Characters
	3	Sending Application
	4	Sending Facility
	5	Receiving Application
	6	Receiving Facility
	7	Date/Time of Message
	8	Security
	9	Message Type
	10	Message Control ID
	11	Processing ID
	12	Version ID
OBR	4	Universal Service Ident.
	7	Observation Date/Time
	8	Observation End Date/Time
	9	Collection Volume
	14	Specimen Received Date/Time
	16	Ordering Provider
	18	Placers Field #1 (Ward/Clinic)
	20	Fillers Field #1 (Ward/Clinic)
	22	Results Rpt/Status Chng-Date/Time
OBX	2	Value Type
	3	Observation Identifier
	5	Observation Results
ORC	1	Order Control
	9	Date/Time of Transaction
	14	Call Back Phone Number
PID	3	Patient ID (Internal ID)
	5	Patient Name
	7	Date of Birth
	8	Sex
	19	SSN Number - Patient

3.5 Fields

SEGMENT	FIELD SEQUENCE NUMBER	FIELD ELEMENT NAME
QRD	1	Query Date/Time
	2	Query Format Code
	3	Query Priority
	4	Query ID
	7	Quantity Limited Request
	8	Who Subject Filter
	9	What Subject Filter
	10	What Department Data Code

4. TRANSACTION SPECIFICATIONS

4.1 General

The flow of transactions between the DHCP Radiology package and the non-DHCP system can occur in one of two ways.

A. DHCP will notify the non-DHCP system that an exam has been done and the non-DHCP system will notify the DHCP system of the results of the exam once the report has been entered.

B. The non-DHCP system will query the DHCP system for an exam list for a patient or for a specific exam and the DHCP system will respond with the appropriate exam information. The non-DHCP system will then send the results of the exam(s) to the DHCP system once the report has been entered.

4.2 Specific Transactions

A. Complete Exam Sent to Non-DHCP System

When an exam is completed on the DHCP system, an Order (ORM) message is sent to the non-DHCP system. The ORM message would consist of the following segments:

ORM	ORDER MESSAGE
MSH	Message Header
PID	Patient Identification
ORC	Common Order
OBR	Observational Request
OBX	Result

4.2 Specific Transactions

EXAMPLE:

```
---------------
MSH^~|\&^RADIOLOGY^608^RADIOLOGY^NON-DHCP^199104301000^^ORM^12345^P^2.1
PID^^^55555~5~M11^^JONES~JOHN~J^^19300101^M^^^^^^^^^^987654321
ORC^NW^^^^^^^^199104301000
OBR^^^^7089898.8453-1~040391-66~L^^^199104301200^""^""^^^^^""^^3232~
HARRIS~JACK^^MEDICINE^^^^199104301000
OBX^^CE^P~PROCEDURE~L^^100~CHEST PA & LAT~L
OBX^^TX^M~MODIFIERS~L^^RIGHT, PORTABLE
OBX^^TX^H~HISTORY~L^^None
OBX^^TX^A~ALLERGIES~L^^BEE STINGS
```

The non-DHCP system then sends a General Acknowledgment (ACK) message back to the DHCP system.

EXAMPLE:

```
---------------
MSH^~|\&^RADIOLOGY^NON-DHCP^RADIOLOGY^608^199104301001^^ORR^54322^P^2.1
MSA^AA^12345
```

B. Results of Exam sent to DHCP System

When the exam results corresponding to the order that was sent by the ORM message in paragraph A are ready, an Observational Results Unsolicited (ORU) message is sent to the DHCP system. The ORU would consist of the following segments:

ORU	OBSERVATIONAL RESULTS UNSOLICITED
MSH	Message Header
PID	Patient Identification
OBR	Observational Request
OBX	Result

EXAMPLE:

```
---------------
MSH^~|\&^RADIOLOGY^NON-DHCP^RADIOLOGY^608^199104301010^ACCESS CODE~~
SIGNATURE CODE^ORU^12346^P^2.1
PID^^^55555~5~M11^^JONES~JOHN~J^^19300101^M^^^^^^^^^^987654321
OBR^^^^7089898.8543-1~043091-66~L^^^199104301200^""^""^^^^^""^^3232~
HARRIS~JACK^^^^MEDICINE^^199104301010
OBX^^TX^I~IMPRESSION~L^^HEART NORMAL SIZE
OBX^^ST^D~DIAGNOSTIC CODE~L^^NORMAL
OBX^^TX^R~REPORT~L^^Heart appears to be of normal size.
OBX^^TX^R~REPORT~L^^No infiltrate or abnormal mass noted.
```

4.2 Specific Transactions

The DHCP system would then send back a General Acknowledgment (ACK) message.

EXAMPLE:

```
---------------
MSH^~|\&^RADIOLOGY^608^RADIOLOGY^NON-DHCP^199104301011^^ACK^54320^P^2.1
MSA^AA^12346
```

C. Query for a List of Exams for a Patient

An alternate method for a non-DHCP system to determine which exams have been completed for a patient is to send a Query Message (QRY) to the DHCP system. The QRY would consist of the following segments:

QRY QUERY MESSAGE

```
----------------------------------------
MSH^~|\&^RADIOLOGY^NON-DHCP SITE^RADIOLOGY^608^199104301100^ACCESS
CODE~~SIGNATURE CODE^QRY^12347^P^2.1
QRD^199104301100^R^I^Q1^^^5~RD^55555^OTH^PATIENT
```

The DHCP system would respond to the query with a list of up to five exams for patient 55555 in record-oriented format. In the following example, only one complete exam existed for the patient.

EXAMPLE:

```
---------------
MSH^~|\&^RADIOLOGY^608^RADIOLOGY^NON-DHCP^199104301101^^ORF^54321^P^2.1
MSA^AA^12347
QRD^199104301101^R^I^Q1^^^1~RD^55555^OTH^PATIENT
PID^^^55555~5~M11^^DOE~JOHN~J^^19300101^M^^^^^^^^^^987654321
OBR^^^^7089898.8453-1~043091-66~L^^^199104301200^""^""^^^^^""^^3232~
HARRIS~JACK^^^^MEDICINE^^199104301200
OBX^^CE^P~PROCEDURE~L^^110~CHEST 1 VIEW~L
OBX^^TX^M~MODIFIERS~L^^RIGHT, PORTABLE
OBX^^TX^H~HISTORY~L^^A history is not available for this patient.
OBX^^TX^A~ALLERGIES~L^^BEE STINGS
```

This query can be used to request a list of exams or just the most recent exam. To request the most recent exam, Field #7 of the QRD segment would specify one record as the quantity (1~RD) in Field #7. To receive a list of exams, more than one record would be specified as in the example above. For either of these queries, the full SSN of the patient or the first letter of the last name and the last four digits of the SSN can be passed as the Who Subject Filter. Likewise, this query can be used to request a specific exam. To do so, Field #7 would specify one record (1~RD), Field #8 would specify the exam number (e.g., 042891-666) or case number (e.g., 666), and Field #10 would specify the word EXAM.

Appendix B. Supported HL7 Message Types

ABBREVIATED NAME:	ACK	FULL NAME:	General Acknowledgment
ABBREVIATED NAME:	ADT	FULL NAME:	ADT Message
ABBREVIATED NAME:	ARD	FULL NAME:	Ancillary Report (Display)
ABBREVIATED NAME:	BAR	FULL NAME:	Add/Change Billing Account
ABBREVIATED NAME:	DFT	FULL NAME:	Detail Financial Transaction
ABBREVIATED NAME:	DSR	FULL NAME:	Display Response
ABBREVIATED NAME:	MCF	FULL NAME:	Delayed Acknowledgment
ABBREVIATED NAME:	OCF	FULL NAME:	Order Confirmation
ABBREVIATED NAME:	ORF	FULL NAME:	Observational Result/Record Response
ABBREVIATED NAME:	ORM	FULL NAME:	Order
ABBREVIATED NAME:	ORR	FULL NAME:	Order Response Message
ABBREVIATED NAME:	ORU	FULL NAME:	Observational Results Unsolicited
ABBREVIATED NAME:	OSQ	FULL NAME:	Order Status Query
ABBREVIATED NAME:	QRY	FULL NAME:	Query
ABBREVIATED NAME:	UDM	FULL NAME:	Unsolicited Display

Appendix C. Supported HL7 Segment Types

ABBREVIATED NAME:	ACC	FULL NAME:	Accident
VERSION: 2.1			
ABBREVIATED NAME:	ADD	FULL NAME:	Addendum
VERSION: 2.1			
ABBREVIATED NAME:	BHS	FULL NAME:	Batch Header
VERSION: 2.1			
ABBREVIATED NAME:	BLG	FULL NAME:	Billing
VERSION: 2.1			
ABBREVIATED NAME:	BTS	FULL NAME:	Batch Trailer
VERSION: 2.1			
ABBREVIATED NAME:	DG1	FULL NAME:	Diagnosis
VERSION: 2.1			
ABBREVIATED NAME:	DSC	FULL NAME:	Continuation Pointer
VERSION: 2.1			
ABBREVIATED NAME:	DSP	FULL NAME:	Display Data
VERSION: 2.1			
ABBREVIATED NAME:	ERR	FULL NAME:	Error
VERSION: 2.1			
ABBREVIATED NAME:	EVN	FULL NAME:	Event Type
VERSION: 2.1			
ABBREVIATED NAME:	FHS	FULL NAME:	File Header
VERSION: 2.1			
ABBREVIATED NAME:	FT1	FULL NAME:	Financial Transaction
VERSION: 2.1			
ABBREVIATED NAME:	FTS	FULL NAME:	File Trailer
VERSION: 2.1			
ABBREVIATED NAME:	GT1	FULL NAME:	Guarantor
VERSION: 2.1			
ABBREVIATED NAME:	IN1	FULL NAME:	Insurance
VERSION: 2.1			
ABBREVIATED NAME:	MRG	FULL NAME:	Merge Patient Information
VERSION: 2.1			

ABBREVIATED NAME: MSA FULL NAME: Message Acknowledgment
VERSION: 2.1

ABBREVIATED NAME: MSH FULL NAME: Message Header
VERSION: 2.1

ABBREVIATED NAME: NCK FULL NAME: System Clock
VERSION: 2.1

ABBREVIATED NAME: NK1 FULL NAME: Next of Kin
VERSION: 2.1

ABBREVIATED NAME: NPU FULL NAME: Non-Patient Update
VERSION: 2.1

ABBREVIATED NAME: NSC FULL NAME: Status Change
VERSION: 2.1

ABBREVIATED NAME: NST FULL NAME: Statistics
VERSION: 2.1

ABBREVIATED NAME: NTE FULL NAME: Notes and Comments
VERSION: 2.1

ABBREVIATED NAME: OBR FULL NAME: Observation Request
VERSION: 2.1
VERSION: 2.2

ABBREVIATED NAME: OBX FULL NAME: Result
VERSION: 2.1

ABBREVIATED NAME: ORC FULL NAME: Common Order
VERSION: 2.1

ABBREVIATED NAME: ORO FULL NAME: Order Other
VERSION: 2.1

ABBREVIATED NAME: PID FULL NAME: Patient Identification
VERSION: 2.1

ABBREVIATED NAME: PR1 FULL NAME: Procedures
VERSION: 2.1

ABBREVIATED NAME: PV1 FULL NAME: Patient Visit
VERSION: 2.1

ABBREVIATED NAME: PV2 FULL NAME: Patient Visit - Additional
VERSION: 2.2

ABBREVIATED NAME: QRD FULL NAME: Query Definition
VERSION: 2.1

ABBREVIATED NAME: QRF FULL NAME: Query Filter
VERSION: 2.1

ABBREVIATED NAME: RX1 FULL NAME: Pharmacy Order
VERSION: 2.1

ABBREVIATED NAME: UB1 FULL NAME: UB82 Data
VERSION: 2.1

ABBREVIATED NAME: URD FULL NAME: Results/Update Definition
VERSION: 2.1

ABBREVIATED NAME: URS FULL NAME: Unsolicited Selection
VERSION: 2.1

ABBREVIATED NAME: ZCT FULL NAME: VA Emergency Contact
VERSION: 2.1

ABBREVIATED NAME: ZDP FULL NAME: VA Dependent Information
VERSION: 2.1

ABBREVIATED NAME: ZEL FULL NAME: VA Patient Eligibility
VERSION: 2.1

ABBREVIATED NAME: ZEM FULL NAME: VA Employment Information
VERSION: 2.1

ABBREVIATED NAME: ZGD FULL NAME: VA Guardian
VERSION: 2.1

ABBREVIATED NAME: ZIC FULL NAME: VA Patient Income
VERSION: 2.1

ABBREVIATED NAME: ZIO FULL NAME: VA Specific Patient Information

ABBREVIATED NAME: ZIR FULL NAME: VA Specific Income Information

ABBREVIATED NAME: ZIV FULL NAME: VA Message Processing
VERSION: 2.1

ABBREVIATED NAME: ZMT FULL NAME: VA Means Test Information
VERSION: 2.1

ABBREVIATED NAME: ZPD FULL NAME: VA Patient Information
VERSION: 2.1

ABBREVIATED NAME: ZTA FULL NAME: VA Temporary Address
VERSION: 2.1

Appendix D. HL7/VA FileMan Data Types

The following table is a list of HL7 data types and their corresponding VA FileMan data types. Listed under the first column titled "Function Call" are entry points in the HLFNC routine that can be called to convert VA FileMan data to HL7 format. Listed under the second column titled "Function Call" are entry points that can be called to convert HL7 data to VA FileMan format.

HL7 Data Type	Function Call	VA FileMan Data Type	Function Call
(ST) String	None Needed	Free Text	None Needed
(TX) Text	None Needed	Word Processing	None Needed
(FT) Formatted Text (See Note below)	None	No Equivalent	None
(NM) Numeric	None Needed	Numeric	None Needed
(DT) Date	HLDATE	Date Only	FMDATE
(TM) Time	HLDATE	Time Only	FMDATE
(TS) Time Stamp	HLDATE	Date/Time	FMDATE
(PN) Person Name	HLNAME	Free Text	FMNAME
(TN) Telephone No.	HLPHONE	Free Text	None Needed
(AD) Address	HLADDR	Free Text	None Needed
(ID) Coded Value	None Needed	Set of Codes or Pointer	None Needed
(SI) Sequence ID	None Needed	Numeric	None Needed
(CM) Composite	None Needed	No Equivalent	None Needed
(CE) Coded Element	None Needed	No Equivalent	None Needed
(CF) Coded Element with Formatted Value	None Needed	No Equivalent	None Needed
(CK) Composite ID with Check Digit	None Needed	No Equivalent	None Needed
(CN) Composite ID and Name	None Needed	No Equivalent	None Needed
(CQ) Composite Quantity with Units	None Needed	No Equivalent	None Needed
(MO) Money	None Needed	No Equivalent	None Needed
(RP) Reference Pointer	None Needed	No Equivalent	None Needed
(TQ) Timing Quantity	None Needed	No Equivalent	None Needed

NOTE: The formatted text (FT) data type is not required to be used in any HL7 fields at this time. The formatted text data type includes formatting instructions (escape codes) in the data string. It is recommended that locally created fields not be assigned the formatted data type. Use the string (ST) or text (TX) data types instead.

Appendix E. HL7 Batch Protocol

The HL7 protocol includes a batch protocol for sending multiple HL7 messages at one time. If a DHCP application is using the MailMan lower level protocol, the use of the HL7 batch protocol is supported. The batch protocol must be implemented in the following manner.

Fields 1 through 8 in the Batch Header (BHS) segment match fields 1 through 8 in the Message Header (MSH) segment. These fields should be created in the BHS segment in the same manner they would be created for the MSH segment.

Field #9, Batch Name/ID/Type must be set equal to four components: batch name (optional), processing ID (required), message type (required), and version ID (required). These four components must be separated by the HL7 component separator. The processing ID, message type, and version ID components correspond to the fields of the same name in the MSH segment.

Field #10, Batch Comment, must be set equal to two components if this batch message is an acknowledgment message. The two components are: acknowledgment code (required) and text message (optional). These two components must be separated by the HL7 component separator, and are equivalent to the ACKNOWLEDGMENT CODE and TEXT MESSAGE fields in an MSA segment.

Field #12, Reference Batch Control ID, must be set equal to the batch control ID being acknowledged, and is equivalent to the MESSAGE CONTROL ID field in an MSA segment.

Including this additional data in the BHS segment will allow the DHCP HL7 package to automatically update status and other information in the HL7 MESSAGE TEXT file (#772) for batch type HL7 transmissions.

Appendix F. Security

The DHCP HL7 package supports the use of the SECURITY field (#8) of the Message Header (MSH) or Batch Header (BHS) HL7 segments. Encryption of the values in this field is not supported in this version of the DHCP HL7 package, but will be supported in the future. DHCP applications might choose not to use the SECURITY field (#8) at all. Those applications that wish to use Security for their HL7 messages must implement the SECURITY field (#8) in the following manner.

The SECURITY field (#8) is divided into three components. The three components must be separated by the HL7 component separator. Only the first and third components are currently used. The second component is reserved for future use.

The first component is required. It must be a valid access code of a user in the NEW PERSON file (#200) on the DHCP system that receives the HL7 message.

The third component is optional. If used, it must be the valid electronic signature code in the NEW PERSON file (#200) on the DHCP system that receives the HL7 message for the user identified by the access code in the first component of the SECURITY field (#8).

Index

Appendix A. HL7 Interface Specifications 51
Appendix B. Supported HL7 Message Types 57
Appendix C. Supported HL7 Segment Types 59
Appendix D. HL7/VA FileMan Data Types 63
Appendix E. HL7 Batch Protocol 65
Appendix F. Security 67

DHCP Interface Guide 9
Developing an Interface Specification 10

Entry Points 43

General Overview 9
Glossary 49

Introduction 1

Lower Level Protocols 3

Organization of this Manual 5
Overview 1, 10, 14, 23, 31

Package Management 7

Receiving HL7 Messages 23
Reference Guide 37
Related Manuals 6

Sending HL7 Messages 31
Setting Up File Entries 14

The DHCP HL7 Package 2
The DHCP Interface to the HL7 Protocol 4

Variables 37