



**BRITISH
COLUMBIA**

Ministry of Health Services

**Professional and Software Compliance Standards
For HL7 Messaging**

**Volume 5 – Network Transmission
and Responses**

Version 3.0

November 21, 2003

Author:	HealthNet
Creation Date:	September 30, 1999
Last Updated:	December 11, 2003
Document Number:	
Version:	3.0

Approvals:

Project Sponsor

Signature

Date

Kathy Hill

HealthNet Access Services

Compliance Process Standards

Approval

Reviewer

Signature:

Signature:

Name:

Kathy Hill

Name:

Robert Grant

Title:

Manager
HealthNet Access Services

Title:

EDI Architect,
Standards & Architecture, IMG

Date:

Date:

October 24, 2003

Contents

1	General Information	4
1.1	What are the volumes in this set?	4
1.2	Corrections and updates	4
1.3	Who is the audience?	4
1.4	Contacting Technical Support	5
1.4.1	Technical Support	5
1.4.2	Help Desk	5
2	Network Transmissions and Responses	6
2.1	Network Input/Output Data Stream	6
2.1.1	Input and Output Transactions – Definition	6
2.1.2	HNSecure Data Stream	7
2.2	Message Structure	7
2.2.1	Overview	7
2.2.2	Message Components	9
2.2.3	Segment Components	12
2.2.4	Data Type Components	13
2.2.5	Table Components	15
2.2.6	Required/Optional Usage Considerations	16
2.2.7	Future Release Considerations	19
2.3	Transaction Construction	19
2.3.1	Message Delimiters	19
2.3.2	Use of Escape Sequences in Text Fields	20
2.3.3	Message Construction Rules	22
2.3.4	Resetting Field Values	23
2.3.5	Specifying Repeating Fields	24
2.4	Transaction Usage	25
2.4.1	MSA Segment	25
2.4.2	Use of Encoding Characters in MSH, MSA, ZHD segments	26
2.4.3	Implied Decimal Points	26
2.4.4	Mixed and Upper Case Data	27
2.4.5	Update Transactions	27
2.4.6	Delete Transactions	27
2.4.7	Time Zones	27
2.4.8	Message Control ID on MSH - Duplicate Transactions	27
2.4.9	Message Type on MSH	28
2.4.10	Encryption Overview	28
2.4.11	Error Response Summary	29
2.5	Transaction Examples	31
2.5.1	HNSecure Timeout – Unable to Connect to HNCLIENT	31
2.5.2	HNSecure Timeout – Unable to Connect to HNGATE/HNGARD	31
2.5.3	R02 - Record New Person	32
2.5.4	R35 - End Payer Relationship	32
2.5.5	R46^Z26 - Update Premium Payment Periods	32
2.5.6	R75^Z62 – Override Pending Registration	33
2.6	No Response From the Network – Timeout & Connection Loss	34

1 General Information

This document and its companion volumes contain the **Professional and Software Compliance Standards for HL7 Messaging** between the BC Ministry of Health and external clients. These standards are used for the exchange of information with various business areas within the Ministry including: the Client Registry (patient/client demographics), MSP (beneficiary coverage), MSP Employer Services (enrolment of employees and dependants), Primary Health Care (patient rostering) and Continuing Care (client demographics and history).

1.1 What are the volumes in this set?

The HL7 Standards for messaging to and from BC Ministry of Health applications are described in a series of business and technical volumes.

Volume 1 – Introduction to the Professional and Software Compliance Standards. A general introduction to the specifications along with tabular listings of all supported messages and message interactions.

Volume 2 – The evaluation process to determine if software is compliant with the Ministry's standards, as described in these documents

Volume 3 – Separate publications containing the business rules for each particular business area. (3a – Client Registry, 3b – MSP Direct are available to date)

Volume 4 – HL7 Message Specifications. A series of standalone documents for each of the transactions used by the BC Ministry of Health.

Volume 5 – Network Transmissions

Volume 6 – Security and Data Integrity

Volume 7 – Glossary

All documentation is available on the HealthnetBC Products and Services Catalogue web site <http://healthnet.hnet.bc.ca/catalogu/tech/compdocs.html>

1.2 Corrections and updates

Corrections and update notes can be found at the end of this document. A vertical line in the outside boarder denotes corrections within the document. ¹

1.3 Who is the audience?

This document is intended for use by:

- a) Software Support Organizations (SSO) who wish to develop software that is compliant with the BC standard for the exchange of Client Registry data and other Ministry supported transactions.
- b) Providers, administrators, health care professionals and MSP Benefits administrators (public and private employers) who are responsible for the implementation of compliant software in their organizations.

1.4 Contacting Technical Support

1.4.1 Technical Support

- a) Software Compliance Standards
- b) Compliance Evaluation Scheduling
- c) Access to software development, testing, production & training databases
- d) Business Development Team
- e) Requests for documentation, bulletins, etc.
- f) Technical support for HNSecure/Infrastructure
- g) Post-implementation support

The Software Support Organization Coordinator is the first point of contact in the Ministry of Health for technical software development support.

Email: HLTH.HnetConnection@gems3.gov.bc.ca

Telephone: 1-250-952-3531

1.4.2 Help Desk

Reference is made in some transaction business rules that participants are required to contact the HealthNet Help Desk to resolve issues. The HealthNet Help Desk line is NOT the first contact point for Vendors.

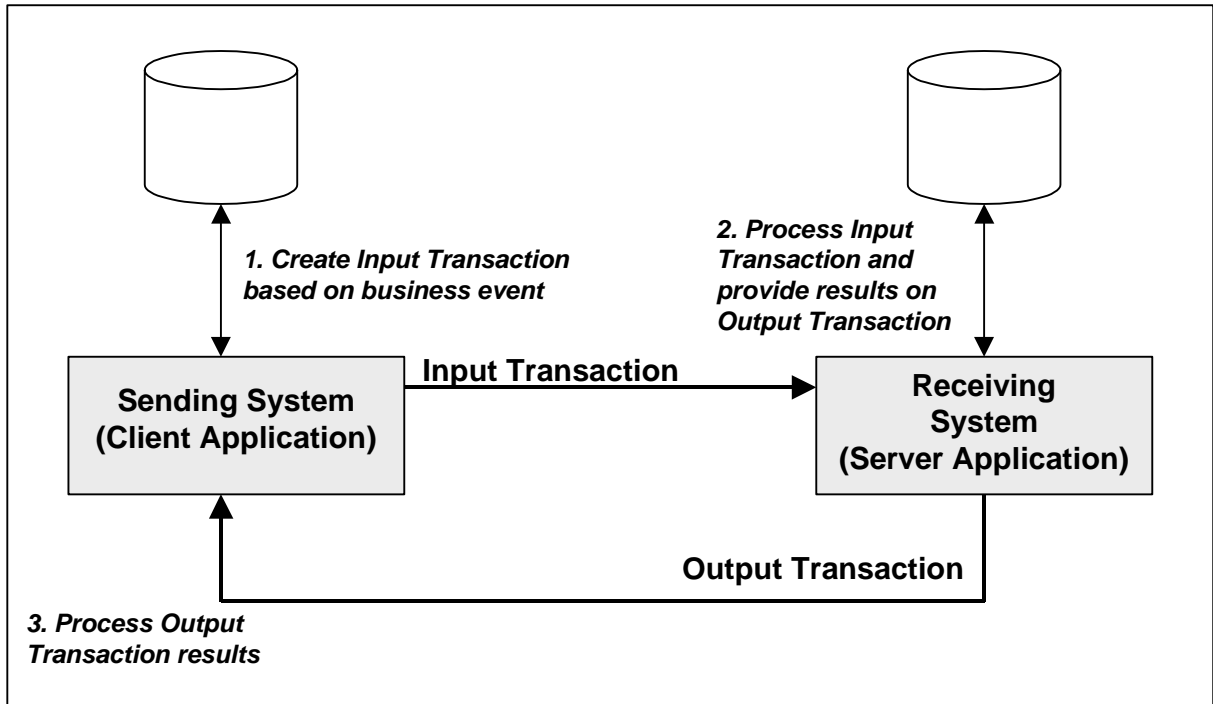
HealthNet Help Desk telephone 1-250-952-1234

2 Network Transmissions and Responses

2.1 Network Input/Output Data Stream

2.1.1 Input and Output Transactions – Definition

The diagram below illustrates the various terms used to document HealthnetBC transactions.



A request is initiated by the Sending System (Client Application) by creating an Input Transaction (e.g. a message requesting a new PHN). This message is then sent to the Receiving System (Server Application) for processing. Once the Receiving System has processed the transaction, it forms a response message (Output Transaction) and sends this back to the Sending System. The Sending System then analyses the Output Transaction and reacts accordingly (e.g. the PHN was created, so update the sending system database records with the new PHN).

Note that this standard often uses Client Application and Sending System software interchangeably, as well as Server Application and Receiving System software. Use of the generic term “application” implies either the Sending System software or Receiving System software.

2.1.2 HNSecure Data Stream

HNSecure does not support the concept of a delivery acknowledgement, therefore the sending system will not be notified that the receiving system has received a request (input transaction). The sending system software must receive and appropriately handle the HNSecure acknowledgements which come in the form of either:

- a valid response from the receiving system (e.g. PHN created); or,
- a standard HL7 ACK reply message (MSH and MSA segments).

The only exceptions to this rule are:

- the sending system software is communicating through HNAPI or through HNLLI and an internal error occurs which makes it impossible to formulate a valid HL7 response. In this case, a non-zero return code is returned.
- the sending system software is communicating directly through HNCLIENT when a network or other serious error occurs which prevents HNCLIENT from returning a response to the sending system software. In this case, there will be a network timeout or the socket connection will be dropped.

2.2 Message Structure

2.2.1 Overview

This section of the document describes how HealthnetBC HL7 (Health Level 7) messages are constructed and documented. The diagram below illustrates the document structure used for HealthnetBC transactions.

HealthnetBC Transaction Segments and Fields – Sending System

PID	Required
-----	----------

Seq		Opt	Reps	Rule	Type	
-	Segment ID	R				"PID"
2	External Patient ID	O		1	HNET :CX	Personal Health Number (PHN) <ul style="list-style-type: none"> ▪ ID (10) - Personal Health Number (PHN); required ▪ Namespace ID of Assigning Authority (5) - Identifier Issuing Jurisdiction; required; valid value is "BC" ▪ Identifier Type Code (2) - required; valid value is "PH"
..						

Appendix A - HealthnetBC Fixed Length Segment Definitions

PID - Person Identification Segment

Seq	Element Name	Len	RP/#	HL7 Data Type	Table	HealthnetBC Usage
-----	--------------	-----	------	---------------	-------	-------------------

Seq	Element Name	Len	RP/#	HL7 Data Type	Table	HealthnetBC Usage
-	Segment ID	3		HL7:SI		"PID"
1	Set ID - PID	4		HL7:SI		
2	External Patient ID	135		HNET:CX		
3	Internal Patient ID	135	0-4	HNET:CX		
...						

Appendix B - HealthnetBC Supported Formats (HL7 Data Types)

HNET:CX - Extended Composite ID with Check Digit

Seq	Component Name	Len	HL7 Data Type	Table	HealthnetBC Usage
1	ID	15	HL7:ST		actual key
2	Check Digit	1	HL7:ST		
3	Code Identifying the Check Digit Scheme Employed	3	HL7:ID	HL7:0061	
4	Assigning Authority	52	HNET:HD		
5	Identifier Type Code	2	HL7:IS	HNET:0203	
...					

Appendix C - HealthnetBC Supported Data Definition Tables

HNET:0203 - Identifier Type Code

Code	Description	HealthnetBC Usage
...		
MH	Ministry of Health Client ID	
PH	Personal Health Number (PHN)	
...		

2.2.2 Message Components

A HealthnetBC transaction is comprised of both an input and output transaction. As per the Health Level 7 (HL7) standard, a message is comprised of 1 or more segments (see example below). Some segments can repeat multiple times (with minimum/maximum repetitions specified) and are either optional or required, depending on the transaction requirements. This document specifies repetition/optional/required for segments as per the example below.

The following specification illustrates the structure of the HL7 message.

{ } denotes one or more repetitions of the enclosed segment(s)

[] denotes that the enclosed segments are optional

Refer to the HealthnetBC Fixed Length Segment Definitions section of this document for more details on each of the segment definitions.

Input Message (R03)

MSH	Message Header	Required
ZHD	HealthnetBC Message Header	Required
PID	Person Identification	Required
[ZIA]	HealthnetBC Person Identification Extension	Optional
[ZSG]	<i>HealthnetBC Digital Signature</i>	<i>Optional, Future Release</i>

Input: In the example above, the ZIA and ZSG segments are optional on the input message - all other segments must be present for this to be a valid transaction.

Output Message (R03) ²

MSH	Message Header	Required
MSA	Message Acknowledgement	Required
[{ ERR }]	Error	Optional, Repeats, minimum 0, maximum 99
[Optional, If person (patient/client) found, then Required
PID	Person Identification	Required
{ ZIA }	HealthnetBC Person Identification Extension	Required, Repeats - minimum 1, maximum 10
]		

Output: In the example above, the ERR segment is optional so may not appear in all instances of the message. It can repeat multiple times (for multiple error messages). Also, the PID and ZIA are both optional, but

required if a person (patient/client) record is found (and there is data to return for this transaction). In addition, the ZIA can repeat if a person (patient/client) record is found, up to a maximum of 10 occurrences of the segment in a message.

Refer to the section Required/Optional Usage Considerations below for more information on required/optional segments, fields and components.

Seq	Field	Req/ Opt	# of Reps	Proc Rule	Data Type	Table	Field Notes
-	Segment ID	R					"PID"
2	External Patient ID	O		1			Personal Health Number (PHN) <ul style="list-style-type: none"> • ID (10) - Personal Health Number (PHN); required • Namespace ID of Assigning Authority (5)- Identifier Issuing Jurisdiction; required; valid value is "BC" • Identifier Type Code (2) - required; valid value is "PH"

Seq

This is the sequence number in the segment. Field 1 is after the 1st field delimiter, field 2 is after the 2nd delimiter, etc. as in |PID||*External Patient ID*|

Field

The HL7 field name (or Element Name) within the segment. This should be referenced in the segment definitions to determine HL7 Data Type (to define format of field) and Table (to define domain/list of values for field). Refer to the Segment Components section below for a description of HL7 Data Type and Table.

Req/Opt

Usage requirements for the field: whether required, optional, conditional or some variant of these three. Refer to the section Required/Optional Usage Considerations below for more information on required/optional segments, fields and components.

of Reps

Number of repetitions for a field specified as **x-y**, where x is the minimum number of field repetitions and y is the maximum number of field repetitions that can be specified. Note that maximum repetitions are bounded by the segment definitions.

Repetitions are separated by the repetition separator (~) as in the following example: |*data~data~data~data*.

Proc Rule

This is a reference number to a processing rule that may impact how this particular field is used in a transaction. For example, there may be 3 identifiers allowed on an input transaction to select a person (patient/client) record. The processing rule may specify that only 1 identifier may be present on input. Therefore, the 3 fields in the transaction would all be optional (not 1 is required every time the transaction is used). The Proc Rule would specify the processing rule # that indicates that 1 identifier must be present on input.

Data Type

The data type for the field, transcribed here from the segment definition in Appendix A – HL7 Fixed Length Segment Definitions for ease of reference. See below for a description of data types.

Table

The vocabulary table used to restrict values for the field, transcribed here from the segment definition in Appendix A – HL7 Fixed Length Segment Definitions for ease of reference. See below for a description of tables.

Field Notes

Field notes describe how the field should be used. Where applicable, the field notes will indicate English names for fields (other than HL7 names) and valid values.

If a field has components (compound field), then the individual components used in the field are identified (with bullet points listing the component or sub-component name such as Identifier Type Code. The components within a field may be required/optional, depending on usage. Refer to the section Required/Optional Usage Considerations below for more information on required/optional segments, fields and components.

For each field and/or component, a maximum length **for the message** is defined and included in brackets; such as (10). **Note that the same field may have a different maximum length in a different message.**

2.2.3 Segment Components

Each segment used is described as in the example below. HealthNet supported segments are described in Appendix A of this publication.

Seq	Element Name	Len	RP/#	HL7 Data Type	Table	HealthnetBC Usage
-	Segment ID	3		HL7:SI		"PID"
1	Set ID - PID	4		HL7:SI		
2	External Patient ID	135		HNET:CX		
3	Internal Patient ID	135	0-4	HNET:CX		
...						

Seq

This is the sequence number in the segment. Field 1 is after the 1st field delimiter, field 2 is after the 2nd delimiter, etc. as in |PID|Set ID|External Patient ID|

Element Name

The HL7 field name (or Element Name) within the segment.

Len

The fixed length for the field. If the field is a component (compound) field, then the length is the total length of the field including all component and sub-component separators (^). The repetition delimiter and repetitions are not included in the Len - that is, the Len is for one repetition of the field. See note on repeating fields for discussion on Len as well as Section 2.6.2 Maximum Length in the HL7 v2.3 specification.

RP#

An indicator if the field can repeat within a field (separated by the repetition separator of ~). If a field can repeat, a value of the form "0-n" will be specified, where n is the maximum number of repeats for the field. Note that the Len of the field is based on the HL7 Data Type and not the number of repetitions. That is, if the HL7 Data Type for HNET:CX is 135 (as above) and the field repeats, the Len will be set to the HL7 Data Type as if only 1 repetition was specified. In the example above for Internal Patient ID, HNET:CX is 135 and the field repeats up to 4 times. Therefore the total length for this field would be (135 * 4) + 3 repeating separators = 543. This document will specify 135 only in the Len field.

HL7 Data Type

The HL7 Data Type will indicate the structure of the field and what kind of data can be included in the field. Data types with a HL7: prefix utilize the HL7 definitions, with the Len specified in the table. Using the example above, fields 1 and 2 are both HL7:SI, but the lengths are 3 and 4 respectively. Data types with a HNET: prefix have been defined by HealthnetBC, typically with a fixed length. For more information on data types, refer to the Data Type Components section below.

Table

The Table will indicate the specific table used to restrict values (domain) for this field. Tables with a HL7: prefix utilize the HL7 definitions, whereas a table with a HNET: prefix have been defined by HealthnetBC. For more information on tables, refer to the Table Components section below.

HealthnetBC Usage

Any notes on how the field should be used/filled/etc. are indicated here. For Z segments, a description of the field is typically included. For non-Z segments, refer to the HL7 documentation for field usage.

2.2.4 Data Type Components

Each HNET: data type used is described as in the example below. HealthNet supported Data Types are described in Appendix B of this publication.

Seq	Component Name	Len	HL7 Data Type	Table	HealthnetBC Usage
1	ID	15	HL7:ST		actual key
2	Check Digit	1	HL7:ST		
3	Code Identifying the Check Digit Scheme Employed	3	HL7:ID	HL7:0061	
4	Assigning Authority	52	HNET:HD		
5	Identifier Type Code	2	HL7:IS	HNET:020 3	
..					

Seq

This is the sequence number in the component field. Component field 1 is after the 1st component field delimiter, component field 2 is after the 2nd component field delimiter, etc. as in |ID^Check Digit^|Code Identifying the Check Digit Scheme Employed^etc.|

Component Name

The HL7 component field name within the component field.

Len

The fixed length for the component field. If the component field is a sub-component (compound) field, then the length is the total length of the component including all sub-component separators (&). For more information, refer to Section 2.6.2 Maximum Length in the HL7 v2.3 specification.

HL7 Data Type

The HL7 Data Type will indicate the structure of the component field and what kind of data can be included in the component field. Data types with a HL7: prefix utilize the HL7 definitions, with the Len specified in the table. Using the example above, component fields 1 and 2 are both HL7:ST, but the lengths are 15 and 1 respectively. Data types with a HNET: prefix have been defined by HealthnetBC, typically with a fixed length. Note that a component field can be broken down into sub-components, such as HNET:HD.

Table

The Table will indicate the specific table used to restrict values (domain) for this component field. Tables with a HL7: prefix utilize the HL7 definitions, whereas a table with a HNET: prefix have been defined by HealthnetBC. For more information on tables, refer to the Table Components section below.

HealthnetBC Usage

Any notes on how the component field should be used/filled/etc. are indicated here. For Z segments, a description of the component field is typically included. For non-Z segments, refer to the HL7 documentation for field usage.

2.2.5 Table Components

Each HNET: table (and some HL7: tables) used in the specifications is described as in the example below. HealthNet supported tables are described in Appendix C of this publication.

Code	Description	HealthnetBC Usage
...		
MH	Ministry of Health Client ID	
PH	Personal Health Number (PHN)	
...		

Code

Actual code value used. A table's values can be restricted by transaction. For example, a table may contain 20 codes, but for a specific transaction, only 1 or 2 are valid. Another transaction may allow a different set of valid codes that can be used, which are different than the 1st set.

Description

English description of the code. Note that the description is not usually returned in a message and is a guideline for displaying coded values to users.

HealthnetBC Usage

Any notes on how the table code should be used/filled/etc. are indicated here. Code values that are planned, but not actually implemented are also noted, as are superseded codes.

2.2.6 Required/Optional Usage Considerations ³

An HL7 message consists of one or more segments, some of which can repeat (like ERR for multiple error transactions). For each of these segments, there are one or more fields. For some of these fields, there are component fields (like Address, which has a street number, street name, city, etc.).

Required and Optional are used to define which segments must or can optionally appear in a message, which fields must or can optionally appear in a segment and which component fields must or can optionally appear in a field.

HealthnetBC has recently (2003) adopted HL7 v2.5 Conformance Usage definitions for fields and components . These enable an exact description to be made of expected application behaviour with respect to these elements (and are therefore preferable to the sometimes ambiguous Required/Optional parameters in the HL7 message specification). The usage codes, which are listed in the table below, are being introduced on new messages only. Existing specifications have not been altered, except in cases where there is demonstrated need to clarify the content.

Usage Codes		
Value	Description	Comment
R	Required	A conforming sending application shall populate all "R" elements with a non-empty value.
RE	Required but may be Empty	The element may be missing from the message, but must be sent by the sending application if there is relevant data. A conforming sending application must be capable of providing all "RE" elements. If the conforming sending application knows the required values for the element, then it must send that element. If the conforming sending application does not know the required values, then that element will be omitted.
O	Optional	HL7 does not permit the use of "O" for implementation specifications. HealthNet will continue to support published specifications which employ the O(ptional) field descriptor. Future publications of implementable specifications will not use "O". Note: "O" usage in HealthNet legacy messages are invariably associated with elements that must be supported by sender/receiver, and which have associated rules governing when the field must be populated. Thus they are equivalent to a usage of "R", "RE", "C", or "CE" and are not truly "optional".
C	Conditional	This usage has an associated condition predicate, testable within the message itself. If the condition is met the element must have a non-empty value.

CE	Conditional but may be empty	<p>This usage also has an associated condition predicate similar to Conditional (C).</p> <p>If the predicate is satisfied: If the conforming sending application knows the required values for the element, then the application must send the element. If the conforming sending application does not know the values required for this element, then the element shall be omitted. The conforming sending application must be capable of knowing the element (when the predicate is true) for all 'CE' elements.</p> <p>If the element is present, the conformant receiving application shall process (display/print/archive/etc.) or ignore the values of that element. If the element is not present, the conformant receiving application shall not raise an error due to the presence or absence of the element.</p> <p>If the predicate is not satisfied: The conformant sending application shall not populate the element. The conformant receiving application may raise an application error if the element is present.</p>
Blank, "-" or "NS"	Not Supported	For conformant sending applications, the element will not be sent. Conformant receiving applications may ignore the element if it is sent, or may raise an application error.

The following table illustrates usage:

PID		Optional			
Seq	Field	Req/ Opt	# of Reps	Proc Rule	Field Notes
-	Segment ID	R			"PID"
2	External Patient ID	O		1	Personal Health Number (PHN) ID (10) - Personal Health Number (PHN); required Namespace ID of Assigning Authority (5) - Identifier Issuing Jurisdiction; required; valid value is "BC" Identifier Type Code (2) - required; valid value is "PH" Namespace ID of Assigning Facility (5) - Identifier Issuing Agency; optional
3	Patient Id (Internal ID)	-			Not Supported
7	Date of Birth	R			CCYYMMDD[HHMMSS][+/-ZZZZ] (19)
8	Sex	R			Gender (1) - valid values are "F", "M", "U"
29	Patient Death Date and Time	RE			CCYYMMDD[HHMMSS][+/-ZZZZ] (19)

In this example, the PID segment is optional and is not required in the message. However, *if* it is included, then the **Req/Opt** column is used to determine which fields in the segment are required. Therefore, if a PID is supplied, the Segment ID, Date of Birth and Sex fields must be included and must not be empty. The Patient Death Date and Time must be included but can be empty if no value is known (e.g. Patient is alive)

Processing Rules may also specify if and/or when a field is required. The **Proc Rule** column indicates a reference to a numbered business rule that may affect how the field is used. Note that the O(ptional) External Patient ID has an associated rule. In more recent formulations this field would instead be marked “RE” (value sometimes not known or not available) or one of “C” or “CE” if the rule referred to some other testable message component.

For Component Fields (those that are made up of components and sub-components - compound fields):

In the example above, the External Patient ID field is optional and is not required in the segment. However, *if* it is included, then the **Field Notes** are used to determine which component fields and sub-component fields in the component field are required. Therefore, if an External Patient ID is supplied, the ID (component), Identifier Type Code (component) and Namespace ID of Assigning Authority (sub-component) must be included. The Namespace ID of Assigning Facility (sub-component) is not required.

Not Supported: Segments and fields not included in the published HealthNet message specification are not supported, meaning that they are ignored by the receiving system. In an actual message instance a “Not Supported” segment should not be present and a ‘Not Supported’ field should be empty.

In exceptional circumstances a field will be published and marked explicitly as “Not Supported”, as in the example above. Field 3, “Patient Id (Internal ID)” is included for information purposes only, and would not be valued in an actual message instance. Typically, these unsupported components are noted in the specification because a) they are required by the HL7 standard but not supported by HealthNetBC services b) they serve to clarify a complex data structure having many components.

2.2.7 Future Release Considerations ⁴

Previous additions of the HealthnetBC compliance specifications included message components marked for future release. With release 2.1 November 21, 2003 these are no longer being published. Should the need arise to alter a message specification it will be re-released either

- as the same message with added optional segments (or segment groups)
- or -
- as an entirely new message.

These measures have been adopted in order to provide for backward compatibility (support for older message types) and to ensure a smooth transition to 'updated' message specifications.

2.3 Transaction Construction

This section of the document indicates how HL7 transactions are constructed and some of the rules associated with the use of HL7. It is by no means complete. The reader is strongly urged to refer to the HL7 v2.3 documentation for more background and additional information/clarity.

2.3.1 Message Delimiters

From HL7 v2.3 documentation, section 2.7 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.

Figure 2-1. Delimiter values

Delimiter	Suggested Value	Encoding Character Position	Usage
Segment Terminator	<cr>	-	Terminates a segment record. This value cannot be changed by implementers.
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 a ST, TX or FT data type, or for use with the data (fourth) component of the ED data type.

Message Delimiter ⁵

Since the HNClient or HNServer is a TCP/IP socket, an end of message is signified when the socket becomes NULL. This is inherent in Socket functionality and there is no requirement for end of message encoding.

2.3.2 Use of Escape Sequences in Text Fields

From HL7 v2.3 documentation, section 2.9 Use of Escape Sequences in Text Fields:

“Formatting codes

When a field of type TX, FT, or CF (or ST - see paragraph below) 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, 0 or more data characters, and another occurrence of the escape character. The following escape sequences are defined:

\H	start highlighting	(not used)
\N	normal text (end highlighting)	(not used)
\F	field separator	
\S	component separator	
\T	subcomponent separator	
\R	repetition separator	
\E	escape character	
\Xddd... \	hexadecimal data	
\Zddd... \	locally defined escape sequence	
\C	carriage return <cr>	(added)

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

No escape sequence may contain a nested escape sequence.

Escape sequences supporting multiple character sets for PN and XPN data types

Not used.

Highlighting

Not used.

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|
^-----^
```

Hexadecimal

When the hexadecimal escape sequence (\Xddd...\) 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 and is beyond the scope of this Standard.

Formatted Text

Not currently used.

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 and is beyond the scope of this Standard.”

2.3.3 Message Construction Rules

From HL7 v2.3 documentation, section 2.10 Message Construction Rules:

“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 2.8, "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 following rules are used, the repetition separator is used only if more than one occurrence is transmitted and 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.

2.3.4 Resetting Field Values

Updating the receiving system's database takes on many forms in HL7. Only fields that have changed in the sending system are sent on an update

transaction. For example, if the sending system performs a retrieve/get person (patient/client) record, then the user proceeds to only update the spelling of the last name, then only the person (patient/client) identifier and last name would appear on the update transaction.

There are 3 methods to update a fields' contents on input (inbound) HL7 transactions:

|data|

sending actual data (e.g. a valid date, coded value, text) will instruct the receiving system to update the contents of its database for that field, assuming a valid transaction.

||

sending 2 vertical bars (field separators) with no data or spaces between the separators will instruct the receiving system to ignore the field. On an update, it indicates that the field has not changed value.

|""|

sending 2 double quotes between 2 vertical bars (field separators) will instruct the receiving system to reset the contents of its database for that field, assuming a valid transaction. This can result in the field either being set to blank, NULL, zero or some other default value (e.g. 1, default code value).

Refer to Transaction Construction Rules section above for more information on use of the "".

2.3.5 Specifying Repeating Fields

If a field allows repetitions, the segment definition (see Appendix A - HealthnetBC Fixed Length Segment Definitions) will indicate the maximum number of repetitions using the format 0-n, where n is a number greater than 0. If there are more than 1 repetition and there are less repetitions to specify in a given transaction, then the repetitions in the field are filled in repetition 1, then repetition 2, etc.

For example, if the field is specified as 2-5 (allows up to 5 repetitions), but the transaction requires 2 repetitions, then they must be specified as |repetition 1~repetition 2|. This is the only valid method of specifying these 2 repetitions. Coding |repetition 1~~~repetition 2| is not valid.

Refer to Message Construction Rules section above for more information on use of the repeating fields.

2.4 Transaction Usage

2.4.1 MSA Segment

At a minimum, each HealthnetBC transaction will return an HL7 ACK reply message consisting of a header segment (MSH) + message acknowledgement segment (MSA), except if an internal error occurred at the sending connections, in which case a non-zero return code is returned to the calling program. Refer to the *HealthnetBC Application Services (HNSecure), Client Functional Specifications, HNAPI/HNCLIENT* document for more information on error handling. See “Available Documentation” in Volume 2.

The sending system may optionally receive additional error, warning and/or information messages in ERR segments if application level (i.e. not infrastructure) messages are returned. Input data application segments may or may not be returned, depending on the rules of the individual transaction. It is the sending system’s responsibility to maintain input parameters and match them to output/response transactions.

The MSA segment will contain the following key fields:

- Acknowledgement Code

This is a summary value representing how the transaction was processed. The following guidelines are to be used for setting and using the value of this field:

AA Application Accept

The transaction was received by the receiving system and processed without application level errors. For example, a lab test was created or a person record updated successfully.

Only Warning and Information messages should be contained in any response transaction (error code in Text Message of MSA or Error Code and Location in any ERR segments).

AE Application Error

The transaction was received by the receiving system and processed with one or more application level errors. For example, a person (patient/client) record was not updated as a required field was missing or not in the list of valid values.

AR Application Reject

This represents a system level error and can be the result of:

- A timeout on the inbound transaction on its way to the receiving system;
- A timeout on the outbound transaction, after processing by the receiving system and before it returns to the sending system;
- Connection errors such as connecting to a remote database;
- Invalid formatted HL7 errors such as missing segments/fields or too many segments;
- Other related system level errors.

In case of timeouts for update transactions, special consideration must be made to determine if the receiving system has actually been updated. If the timeout occurred on the outbound transaction, then the receiving system may or may not have been updated. It is up to the sending system to determine if the receiving system was updated by verifying the state of the receiving system and potentially resubmitting the update transaction.

- **Text Message**

This field is typically defined with an error code (8 bytes) + text message (72 bytes), although individual transaction specification may override this definition. Examples are:

“RECVO22E Connection to HNCLIENT timed out with no or partial response.”

“LOLT001ILab Test created”

2.4.2 Use of Encoding Characters in MSH, MSA, ZHD segments

The use of standard HL7 encoding characters in MSH, MSA and ZHD is discouraged, as this may cause problems with HL7 parsing software. These encoding characters are: “|”, “^”, “~”, “\” and “&”. For example, the Security field on MSH of “SMITH & WESSON” contains the “&” encoding character. If this is a common situation with any field on these segments, contact the HealthnetBC SSO Coordinator (see Volume 1) for further direction.

2.4.3 Implied Decimal Points

HealthnetBC transactions follow HL7 conventions and do **not** use implied decimal points. If a field requires decimal points (e.g. a dollar amount), then the Field Notes for the field/segment in the transaction OR the HealthnetBC Usage for the field in the fixed length representation of the segment will specify the format mask for the field.

2.4.4 Mixed and Upper Case Data

HealthnetBC HL7 transactions are by default upper case, unless noted otherwise in the Field Notes for a particular field for a specific transaction. An example of a mixed case field is the Comments field in the NTE segment. By transmitting data in upper case, it does not preclude the data from being stored in mixed or lower case. For example, coded values (and likely names) must be transmitted in upper case, but may be stored in the database in mixed case.

2.4.5 Update Transactions

All HealthnetBC update transactions must contain only the application data fields that have changed in the sending system and appropriate key(s) to uniquely identify the record requiring update. For example, if a person's name was updated, the update message would only contain the person key + changed name (plus other information required to confirm userid, who sent the transaction, etc.). However, no other application data would be included in the message unless the transaction specifically identifies additional data requirements. These update messages must not include default values for fields that the sending system user has not reviewed and/or adjusted.

2.4.6 Delete Transactions

All HealthnetBC delete transactions must be preceded immediately by a Retrieve or Get transaction unless specifically noted otherwise in the Transaction Details (Volume 4 message specifications).

In addition, the sending system must perform a confirmation (e.g. "Are you sure you want to delete this record? - Y/N") prior to sending the delete transaction.

2.4.7 Time Zones

All times noted in HL7 HealthnetBC transactions will assume Pacific Time unless specifically noted in the transaction specifications.

If the transaction is generated from a time zone other than the PST time zone, then the time must be converted to PST for the transaction, unless specifically noted in the transaction specifications. This affects HNET:TS, and HNET:TM data types.

2.4.8 Message Control ID on MSH - Duplicate Transactions

The Message Control ID on MSH, along with Sending Facility and Sending Application on MSH is used to uniquely identify a transaction on the HealthnetBC network.

A receiving system (e.g. Client Registry) **may** or **may not** check for duplicate transactions sent with the same Message Control ID, Sending Facility and Sending Application. It is therefore up to the sending system software to

determine appropriate actions to take in the event of the receiving system or HealthnetBC returning a timeout response. These actions may include confirming that the transaction was or was not processed by the receiving system (e.g. did adding a new person (patient/client) record take effect or does the add transaction need to be resubmitted?). If determination cannot be made through HealthnetBC transactions, then the HealthnetBC Help Desk may be contacted for assistance.

Note that if the receiving system does check for duplicate transactions, and the transaction needs to be resubmitted, the Message Control ID on MSH must be incremented in order to make the transaction unique.

2.4.9 Message Type on MSH

The following message type identifiers/extensions have been reserved by HealthnetBC: ⁶

- ZFG get a file attachment from the specified transaction/file server
- ZFP send an attached file to the specified transaction/file server
- ZPN transaction/response is in a PharmaNet compatible format

2.4.10 Encryption Overview

HNSecure ensures all data, including attached files, in every transaction is encrypted. Each transaction is encrypted using a unique triple DES key.

2.4.11 Error Response Summary ⁷

This section summarizes error response behaviour. It repeats some information found elsewhere in this publication. Suggested additional reading: ERR and MSA segment definitions in Appendix A; HNET:9000 table in Appendix D.

Network Errors

Assuming that HNSecure is able to formulate a response message (see early section on HNSecure Data Stream) at a minimum each HealthnetBC transaction will return an HL7 ACK reply message consisting of a header segment (MSH) + message acknowledgement segment (MSA). The optional ERR segment will normally not be present. Communication errors are returned in MSA.3 An example can be found in section 2.5.1 - HNSecure Timeout.

Receiving System Errors

A valid response from the receiving system may contain error, warning and/or information messages in the optional and repeatable ERR segments that are present on all response messages:

Example of R02 response message structure:

```
MSH
MSA
[ { ERR } ]
[ other segments defined for response message ]
```

In the event of an outright error (as opposed to warning or information messages) the other optional segments, after ERR, are unlikely to be present in the response message.

Application errors are returned in both MSA.3 – Text Message, and in ERR.1.4.1 – Identifier on Code Identifying Error. (See segment definitions in Appendix A). The first error code returned by the application is repeated in these two segments as follows:

- MSA.3 contains a concatenated string consisting of
 - An error code (7) +
 - An error type code (1) +
 - message text (in English) (72)
- the first ERR segment that is returned repeats these components but delimits the text expression as follows:
 - ERR.1.4.1 contains the error code (7) and concatenated type code (1)
 - ERR.1.4.2 contains the message text (72)

Example:

```
MSH|^~\&|RAIRCRD-NW-
RSN|BC0003000|ADT1|NF20|20030717140837|Train96|R02
|1232418|D|2.3
MSA|AE|19980915000020|HNHR517EDUPLICATE CLIENT FOUND ON
DATABASE - CLIENT NOT ADDED
ERR|^^^HNHR517E&DUPLICATE CLIENT FOUND ON DATABASE - CLIENT NOT
ADDED
```

If there are additional application level errors they are listed in successive iterations of the ERR segment. There will always be only one instance of the MSA segment.

Error Code Interpretation. Application response codes for error, warning and information messages are translated by HealthnetBC when a response message is created. The associated text returned in MSA.3 and ERR.1.4.2 is also modified to be readily understandable to a user, and is intended for display purposes.

Application confirmation codes. Confirmation codes for successful transactions are returned in both MSA.3 – Text Message, and in ERR.1.4.1 – Identifier on Code Identifying Error. (See segment definitions in Appendix A). The confirmation coding is repeated in these two segment locations. In the example below the error type code = “I”, indicates that this an Information code (as opposed to Error or Warning).

Example:

```
MSH|^~\&|RAIRCRD-NW-
PRSN|BC0003000|ADT1|NF20|20030717140908|Train96|R02|1232420|D|2.3
MSA|AA|19980915000020|HJMB001ISUCCESSFULLY COMPLETED
ERR|^^^HJMB001I&SUCCESSFULLY COMPLETED
○
○
○
```

Multiple confirmation codes. Some transactions, like the R02 – Record New Person, are completed through the actions of multiple receiving applications, each of which generates a confirmation code. When translated by HealthNet these confirmation codes are returned as multiple occurrences of ERR, with identical coding, as in the example below:

```
MSH|^~\&|RAIRCRD-NW-
PRSN|BC0003000|ADT1|NF20|20030717140908|Train96|R02|1232420|D|2.3
```

```
MSA|AA|19980915000020|HJMB001|SUCCESSFULLY COMPLETED
ERR|^^^HJMB001I&SUCCESSFULLY COMPLETED
ERR|^^^HJMB001I&SUCCESSFULLY COMPLETED
ERR|^^^HJMB001I&SUCCESSFULLY COMPLETED
PID||9865821574^^^BC^PH^MOH
ZIA||||||||||||1210-647 MICHIGAN ST^^^^^^^^^^^^^^^^^^^^VICTORIA^BC^V8V
1S9^CAN^^^^^Y^20030717^^^^^^N
```

2.5 Transaction Examples

2.5.1 HNSecure Timeout – Unable to Connect to HNCLIENT

Sending System [Input] Transaction

```
MSH|^~\&|ADT00001|BC00000098|LTS00001|BC0003000|19980915231202.4512-
0800|JSMITH|ZLI^L01|19980915000001|P|2.3|||||<cr>
{ + remaining data segments for transaction }
```

Receiving System [Output] Transaction

```
MSH|^~\&|HNAPI||ADT00001|BC00000098|19980915231205.3122-
0800|JSMITH|ACK|19980915000001|P|2.3|||||<cr>
MSA|AR|19980915000001|RECV022E Connection to HNCLIENT timed out with no or partial
response.||||
```

Notice that Sending Application (field #3) is set to HNAPI, the component that determined the time out situation. The Sending Facility is set to blank as the transaction did not leave HNAPI and a network facility ID could not be determined.

2.5.2 HNSecure Timeout – Unable to Connect to HNGATE/HNGARD

Sending System [Input] Transaction

```
MSH|^~\&|ADT00001|BC00000098|LTS00001|BC0003000|
19980915231202.4512-0800|JSMITH|ZLI^L01|19980915000001|P|2.3|||||<cr>
{ + remaining data segments for transaction }
```

Receiving System [Output] Transaction

```
MSH|^~\&|HNCLIENT|BC01000018|ADT00001|BC00000098|
19980915231205.3122-0800|JSMITH|ACK|19980915000001|P|2.3|||||<cr>
MSA|AR|19980915000001|TXFR024E Connection with remote facility timed out.||||
```

Notice that Sending Application (field #3) is set to HNCLIENT, the component that determined the time out situation. The Sending Facility is set to the Network Facility ID of HNCLIENT (BC01000018).

2.5.3 R02 - Record New Person

Sending System [Input] Transaction

```
MSH|^~\&|ADT1|NF20|RAIRCRD-NW-PRSN|BC0003000|1999
0531144853|Train96|R02|19980915000020|D|2.3<cr>
ZHD|19990531144853|^00000010|TRAININGAdmin<cr>
ZTL||N<cr>
PID|||||19730515|M<cr>
ZIA|||||DONOGHUE^JAMES^JOE^^^L|1210-647 MICHIGAN ST
^^^^^^^^^^^^^^^^^^^^VICTORIA^BC^V8V1S9^CAN
^H|^PRN^PH^^250^9526579|||||N
```

Receiving System [Output] Transaction

```
MSH|^~\&|RAIRCRD-NW-PRSN|BC0003000|ADT1|NF20|
20030717140908|Train96|R02|1232420|D|2.3<cr>
MSA|AA|19980915000020|HJMB001|SUCCESSFULLY COMPLETED<cr>
ERR|^HJMB001I&SUCCESSFULLY COMPLETED<cr>
ERR|^HJMB001I&SUCCESSFULLY COMPLETED<cr>
ERR|^HJMB001I&SUCCESSFULLY COMPLETED<cr>
PID||9865821574^^^BC^PH^MOH<cr>
ZIA|||||1210-647 MICHIGAN ST^^^^^^^^^^^^^^^^^^^^VICTORIA^BC^V8V
1S9^CAN^^^^^Y^20030717^^^^^N
```

Notice the repetition of ERR. This particular message is completed by 3 back-end applications which are each reporting successful processing.

2.5.4 R35 - End Payer Relationship

Sending System [Input] Transaction

```
MSH|^~\&|HR|BC00000098|RAIEND-PYR-RL|BC0003000|20010719125635|PM3
3283|R35|20010915000017|D|2.3<cr>
ZHD|20010719125635|^00000001|TRAININGAdmin<cr>
PID||9029274643^^^BC^PH<cr>
IN1|||||2100030|||||20030531<cr>
ZIH|||||K
```

Receiving System [Output] Transaction

```
MSH|^~\&|RAIEND-PYR-RL|BC0003000|HR|BC00000098|2003
0718130825|PM33283|R35|1232626|D|2.3<cr>
MSA|AA|20010915000017|HJMB001|SUCCESSFULLY COMPLETED<cr>
ERR|^HJMB001I&SUCCESSFULLY COMPLETED
```

2.5.5 R46^Z26 - Update Premium Payment Periods

Sending System [Input] Transaction

```
MSH|^~\&|HR|BC00000098|RAICOVPART-ENQ|BC0003000|19
```



```
990719102151|PM33283|R46^Z26|19980915000020|D|2.3<cr>
ZHD|19990719102151|^00000010|TRAININGAdmin<cr>
PID||9076873472^^^BC^PH<cr>
IN1||||||6113872<cr>
ZIN||20020731||D<cr>
ZIN||20020831||U
```

Receiving System [Output] Transaction

```
MSH|^~\&|RAICOVPART-
ENQ|BC0003000|HR|BC00000098|20030718131224|PM33283|R46^Z26|1232628|D|2.3<cr>
MSA|AA|19980915000020|HJMB001|SUCCESSFULLY COMPLETED<cr>
ERR|^HJMB001I&SUCCESSFULLY COMPLETED
```

2.5.6 R75^Z62 – Override Pending Registration

Receiving System [Output] Transaction

```
MSH|^~\&|HNWeb|BC01000106|PHCPEND-OVRD|BC00002043|
20020801105005|uma|R75^Z62|20020801105005|D|2.3<cr>
ZHD|20020801105005|^00000010|phcuser<cr>
PID||9013069354^^^BC^PH<cr>
ZPY||ORG|^10879^^^BC^XX^MOH<cr>
EVN|Z58|20020801105005|04<cr>
ZIL|PHC|||||N<cr>
NTE|||Comments20030618 Gary
```

Receiving System [Output] Transaction

```
MSH|^~\&|PHCPEND-OVRD|BC00002043|HNWeb|BC0
1000106|20030718135033|uma|ACK^Z62|1232638|D|2.3<cr>
MSA|AA|20020801105005|SUCCESS SUCCESS <cr>
ERR|^SUCCESS &SUCCESS <cr>
ERR|^HJMB001I&SUCCESSFULLY COMPLETED
```

2.6 No Response From the Network – Timeout & Connection Loss

Transactions using HNSecure as supplied by the Ministry are subject to a transaction timeout value that is a configurable parameter. It is recommended that the timeout value be set to no less than 70 seconds.

For network errors, including timeouts and connection loss, HNSecure will create an ACK reply message consisting of a single MSH segment + one MSA segment⁸. The following fields are set specifically by HNSecure and must be used to determine if a network error occurred:

- Message Type on MSH = “ACK”
Note: in the case of timeouts, the value of “ACK” will always be set for Message Type.
- Sending Application on MSH = “HNAPI”, “HNCLIENT” or “HNGATE”, depending on where the timeout occurred.
- Sending Facility on MSH = network facility ID of HNCLIENT or HNGATE, depending on where the timeout occurred.

In addition, the Text Message on MSA will be set to indicate the network error condition.

In the case of connection loss, HNClient should make use of alternate connections to the target HNGARD and HNGATE if available.



DOCUMENT MODIFICATION HISTORY		
VERSION	RELEASE DATE	DESCRIPTION
2.0	September 1999	Original single document
3.0	November 21, 2003	<ul style="list-style-type: none"> • Revised publication format. Minor text corrections and updated references • New component usage definitions added to section 2.2.6 • Amended policy on “Future Release” components, section 2.27 • New section 2.4.11 added that summarizes error response behaviour.

Corrections and Update Notes

¹ 02/Nov/27 – example of correction

² Corrections to illustration. The ERR segment shown in the example is Optional, as indicated by the square brackets, with a minimum cardinality of 0.

³ Field Usage code set expanded for recent message specifications. Text and new reference table added.

⁴ Policy change on “Future Release” components, as described in text.

⁵ Added new technical note on message delimiting. Not published in version 2.0.

⁶ These HealthNet specialized message types (from version 2.0 of the standard) were not implemented, but are reserved for future use. Details will be published if and when support is offered for these services.

⁷ New section.

⁸ Text clarification added to notes on network timeout response message.