

# **The Hitchhiker's Guide to NAHLN Messaging**

## **Don't Panic**

Michael K Martin, DVM, MPH  
Editor and Chief Scribe

With Contributions From The NAHLN IT Committee

## Table of Contents

|  |    |
|--|----|
| Table of Contents .....                        | i  |
| Introduction .....                             | 1  |
| Overview .....                                 | 2  |
| What is the NAHLN? .....                       | 2  |
| History: .....                                 | 2  |
| NAHLN messaging: .....                         | 2  |
| Benefits of NAHLN messaging .....              | 3  |
| System Architecture .....                      | 4  |
| Messaging Standards .....                      | 4  |
| Data Standards .....                           | 5  |
| Security Standards .....                       | 6  |
| Requirements for participation .....           | 7  |
| Laboratory Information System Capability ..... | 7  |
| Network Capability .....                       | 9  |
| Standard Terminology Support .....             | 9  |
| NAHLN System Account .....                     | 10 |
| What to send and when to send it? .....        | 10 |
| Supported diseases and tests .....             | 11 |
| Message transmission trigger .....             | 12 |
| Tools of the Trade .....                       | 12 |
| Validating XML Editor .....                    | 13 |
| Schema Files .....                             | 14 |
| NAHLN Result Message description .....         | 15 |
| Abstract Message Definition .....              | 15 |
| Observation Population Unsolicited (OPU) ..... | 16 |
| Message Header Information .....               | 18 |
| Addresses: .....                               | 18 |
| Message Control ID: .....                      | 18 |
| Processing ID: .....                           | 18 |
| Message Profile Identifier: .....              | 18 |
| Visit (Accession) Information .....            | 19 |
| Related Parties Information .....              | 20 |
| Accession Detail .....                         | 20 |
| Animal Owner Information .....                 | 21 |
| Animal Information .....                       | 22 |
| Age Categories .....                           | 23 |
| Specimen Information .....                     | 24 |
| Order Information .....                        | 25 |
| Common Order Information .....                 | 25 |
| Result Information .....                       | 26 |
| NAHLN Order Message Description .....          | 27 |
| Abstract Message Definition .....              | 28 |
| Message Header Information .....               | 28 |
| Role Segments .....                            | 29 |

## The Hitchhiker's Guide to NAHLN Messaging

|  |    |
|--|----|
| ORDER Group.....                                       | 29 |
| SPECIMEN Group.....                                    | 29 |
| OBSERVATION REQUEST Group .....                        | 30 |
| TIMING Group.....                                      | 30 |
| Keeping Similar NAHLN messaging Concepts Straight..... | 30 |
| Addresses .....  | 32 |
| Times.....   | 33 |
| Statuses .....   | 36 |
| Identifiers .....                                      | 37 |
| How are they assigned? .....                           | 37 |
| Who is responsible to assign them? .....               | 39 |
| Hierarchic Designators (HD) .....                      | 40 |
| Key Identifiers and Why We Need Them.....              | 41 |
| Actors and Roles .....                                 | 43 |
| Responsible Diagnostician.....                         | 44 |
| Submitter.....   | 44 |
| Collector.....   | 45 |
| Attending Veterinarian.....                            | 45 |
| Payor .....  | 46 |
| Premises Role (An Odd Variation) .....                 | 46 |
| Data Enterer and Verifier.....                         | 48 |
| Owner.....   | 48 |
| Patients, Samples, Specimens .....                     | 49 |
| Unique Specimen Identifiers.....                       | 50 |
| Program Identifiers .....                              | 52 |
| Observations .....                                     | 54 |
| Patient History .....                                  | 55 |
| Physical Findings.....                                 | 55 |
| Additional Information About the Specimen.....         | 55 |
| Results.....   | 55 |
| Reason for Study.....                                  | 56 |
| Parents.....   | 56 |
| Case Numbers .....                                     | 58 |
| Coded Entries.....                                     | 58 |
| Pooling.....   | 59 |
| Flavors of NULL.....                                   | 61 |
| Sending Nothing .....                                  | 64 |
| Lab-to-Lab Order Messaging.....                        | 65 |
| Redirection of Specimens .....                         | 65 |
| Confirmation Testing.....                              | 65 |
| Compromises and Less-Than-Ideal Solutions .....        | 66 |
| Cycle Thresholds .....                                 | 66 |
| Result Interpretation, AKA “Abnormal Flags” .....      | 67 |
| Non-results and Qualified Results .....                | 67 |
| Test Interpretation vs. Case Classification.....       | 69 |
| Repeated Testing or Re-Sent Results.....               | 70 |

## The Hitchhiker's Guide to NAHLN Messaging

|  |     |
|--|-----|
| Changes to "Orders" .....                                    | 70  |
| NAHLN Standard terminology .....                             | 71  |
| LOINC .....  | 73  |
| SNOMED.....  | 75  |
| Getting Help.....  | 79  |
| HL7 defined terminology.....                                 | 80  |
| NAHLN Defined Terminology.....                               | 80  |
| Terminology Updates.....                                     | 81  |
| Terminology Web Service .....                                | 82  |
| Using the NAHLN Terminology Service .....                    | 83  |
| Message Assembly.....  | 84  |
| How Do You Know Your Message is "Correct?" .....             | 85  |
| Message Delivery.....  | 86  |
| Protocols .....  | 86  |
| Identification and Authentication Protocols .....            | 87  |
| Role-based Access Control .....                              | 88  |
| Low Level Protocols .....                                    | 89  |
| Hypertext Transport Protocol .....                           | 89  |
| Transport Layer Security .....                               | 90  |
| Hypertext Transport Protocol Secure.....                     | 92  |
| Accounts and Passwords .....                                 | 92  |
| Requesting Accounts .....                                    | 92  |
| Updating Passwords.....                                      | 93  |
| Getting Help.....  | 93  |
| Message Transmission Tools .....                             | 93  |
| cURL.....  | 94  |
| Rhapsody.....  | 96  |
| Security Tools .....   | 97  |
| Windows Certificate Manager .....                            | 98  |
| Mozilla Firefox .....  | 99  |
| Internet Explorer .....                                      | 100 |
| Configuring Rhapsody to use Trusted (SSL) Certificates ..... | 100 |
| Configuring cURL to Use vsapps Certificate .....             | 102 |
| Java Keytool.....  | 102 |
| Dealing with Acknowledgements .....                          | 102 |
| Deprecation Warnings .....                                   | 104 |
| Acknowledgements When Using a Message Router .....           | 104 |
| Messaging Details for Specific Test Programs .....           | 105 |
| Wildlife Services Avian Influenza.....                       | 105 |
| Influenza A Virus in Swine (IAV-S) Surveillance .....        | 108 |
| The Premises "Role" .....                                    | 108 |
| Age Estimates .....  | 110 |
| Specimen Barcodes .....                                      | 111 |
| Parent Specimen.....   | 111 |
| Order Group for Entire Protocol .....                        | 112 |
| Reason for Submission .....                                  | 112 |

## The Hitchhiker's Guide to NAHLN Messaging

|  |     |
|--|-----|
| Order Group for Subtyping PCR .....                      | 112 |
| Sequencing Results .....                                 | 113 |
| Porcine Enteric Coronavirus Testing .....                | 113 |
| Conventional vs. Real-time PCR .....                     | 114 |
| Age Categories .....                                     | 114 |
| Unique Specimen Identifiers .....                        | 116 |
| Environmental Samples .....                              | 117 |
| Bovine Spongiform Encephalopathy (BSE) (Draft) .....     | 117 |
| Scrapie .....  | 118 |
| Specimens in Scrapie .....                               | 121 |
| Non-result and Qualified-result Results in Scrapie ..... | 122 |
| Non-Program Initiated Testing .....                      | 123 |
| Premises Role .....                                      | 124 |
| Submitter Role .....                                     | 124 |
| PATIENT Group .....                                      | 125 |
| Specimen ID .....  | 126 |
| Purpose of Tests .....                                   | 127 |
| Conclusion .....   | 127 |
| Acknowledgements .....                                   | 127 |
| Notices .....  | 128 |
| Future Topics .....                                      | 128 |

## Introduction

In *The Hitchhiker's Guide to the Galaxy*<sup>1</sup>, Douglas Adams provides us with some very profound guidance in the form of some pretty insane story telling. The book's name comes from a fictional electronic book by the same name. The fictional book provides guidance for any poor souls who find themselves wandering the universe after a minor explosion destroys their home planet. *The Hitchhiker's Guide to NAHLN Messaging* tries to provide similar guidance to those just venturing out from their usual orbits to explore the exciting but often intimidating galaxy of Veterinary Informatics Standards that make up the National Animal Health Laboratory Network (NAHLN) messaging.

This guide is designed to provide an overview of the concepts of NAHLN messaging—and now all Laboratory Messaging System messaging—in relatively plain English. NAHLN messaging is by its nature a very technical topic. Instead of guiding a user through the steps needed to enter data into a user interface, this guide aims to help with the process of designing a computer system to build and send NAHLN messages directly from existing Laboratory Information Systems. Designing and building such a system will, of course, involve the lab's computer technicians. But the biggest challenge in a NAHLN messaging project is getting the information content right; and laboratory diagnosticians are the ones who know this material. This guide is therefore intended for both audiences.

Many NAHLN laboratories are accustomed to using their information systems for very narrowly focused, workflow-oriented functions. They are now venturing from this to creating universally understood messages capable of flowing through the NAHLN to various surveillance and management systems. This transition may feel a little like hailing a passing flying saucer to escape an exploding planet. *The Hitchhiker's Guide* offers some valuable advice to these travelers. First, always bring your towel. “A towel is the most massively useful thing an interstellar hitchhiker can have.” But even before it provides this suggestion, the guide provides its most important advice. On the cover, in large friendly letters, it contains the words, “Don't Panic.”

These are important words to remember when starting a NAHLN messaging project. The Veterinary Informatics technology used in the NAHLN has been developed over literally decades. One can—and some do—get a PhD in this stuff and spend an entire career studying it. No one is capable of fully understanding all the theory behind every part of the system. But “Don't Panic.” You don't have to do it all yourself any more than you have to be an automotive engineer to drive a car. All you need to do is to understand a few underlying principles and put your faith in the scientists and engineers who have built the tools. This book, and the live training sessions that will be provided, will give you the basic principles and the tools to connect to the NAHLN. Along the way you will get a little peek into the field of Veterinary Informatics and meet some of those who

---

<sup>1</sup> If you are unfamiliar with Douglas Adams' original, it is suggested that you read it. You will get the jokes in here better that way. On the other hand, the actual content of this guide is almost but not completely unrelated to the original, especially in the quality of the writing and the humor.

study it and do the interesting, but sometimes intimidating, work of developing these standards.

## **Overview**

### ***What is the NAHLN?***

#### **History:**

Animal disease surveillance functions most effectively as a shared responsibility of publicly funded state animal health laboratories, represented by the American Association of Veterinary Laboratory Diagnosticians (AAVLD), and federal animal health laboratories administered through the United States Department of Agriculture (USDA) Animal and Plant Health Inspection Service (APHIS). The basic infrastructure of a national laboratory network provides critical features including:

- A secure communication, reporting and alert system
- Standardized, rapid diagnostic techniques that can be used at the state, regional and national level
- Modernized equipment and experienced personnel trained in the detection of emergent, foreign and bioterrorist agents
- A national training, proficiency testing and quality assurance system to ensure that all laboratories in the system meet quality standards
- Federal and state facility upgrades to meet biocontainment requirements
- Periodic scenario testing of the network and the associated response network

In 2002, the USDA established a pilot National Animal Health Laboratory Network (NAHLN) that included animal health laboratories in twelve states. In 2004, the USDA announced the expansion of the NAHLN to include all labs that currently had a diagnostic testing contract with the USDA. This included labs testing for program diseases (e.g. Brucellosis and Pseudorabies), prion diseases, Exotic Newcastle Disease, and Avian Influenza and represented a major expansion of the NAHLN.

Standardized NAHLN messaging was first piloted in March of 2004 with the initial laboratory results message. Based on experiences with the pilot, a second release was produced in early 2005, with additional minor modifications made as needed to support additional diseases and functionality. This release demonstrated the need for additional features. These features were included in the message structure that is currently in use and is discussed in this document.

#### **NAHLN messaging:**

In order to provide, “A secure communication, reporting and alert system” the NAHLN includes a system by which results can be sent over a network to a central database for use by surveillance systems, etc. To be useful, the data in these messages must be consistent between laboratories. But the member laboratories use a wide range of laboratory information systems and local vocabulary systems and codes. It would be completely inappropriate for the NAHLN to force all labs to use the same software or

vocabularies. Instead, the NAHLN specifies a number of standards that are used to translate the local data into common format for transmission.

In order to successfully send and receive messages using the NAHLN system, a laboratory must have its data in unambiguous electronic format. Generally, this means the data are in a Laboratory Information Management System (LIMS) using some sort of controlled list of terms and a structured database. The lab must be able to access these data in order to extract the needed information, map it to the standard format and code systems, and send as messages. Reliable network connectivity to the Internet is also a basic requirement. It is helpful if the LIMS provides a mechanism for “triggering” the building and sending of messages with or without a human approval step.

Beyond these basic requirements, all one needs to participate in NAHLN messaging is the interest, motivation, and will to work through the process of understanding the data and building the necessary translations. The rest of this manual will provide the concepts and suggested procedures for this process. Details of the message construction are available in the Implementation Guide(s) and program-specific messaging guides. Details of the various vocabularies as well as tools to assist in mapping are available on-line through the NAHLN Vocabulary Service (<http://vtsl.vetmed.vt.edu/nahln/main.cfm>).

The first NAHLN message to be defined communicates the results of tests to the repository. But this is just the first of a series of related messages that will ultimately include laboratory orders—so your lab doesn't have to do all the data entry!—and referral messages. Because the NAHLN messaging is built upon well-established Veterinary and Medical Informatics standards, the technology, mappings, and structures used in the NAHLN will be valuable in possible future networking efforts. Examples already in the works include the Food Emergency Response Network (FERN), and the CDC's Electronic Laboratory Reporting (ELR) protocol. So your investment in learning and applying the NAHLN messaging technology is likely to pay ongoing dividends.

### Note on Outbreak Detection

An outbreak is by definition the incidence of a disease above the normally expected level. For Foreign Animal Diseases (FADs), like highly pathogenic avian influenza or foot and mouth disease, the expected incidence is zero and a single case is therefore an outbreak. A system to aggregate results is not needed for detection of such an event but will be useful in monitoring and management.

For diseases with low but non-zero expected incidence, an outbreak may begin with levels too low to be recognized by any single laboratory but that will be apparent when these results are aggregated at a national level. Such aggregation is only possible if data are sent in consistent structure and content.

The NAHLN messaging system can support both detection of outbreaks in low level endemic diseases, and monitoring and management of FADs should they occur.

### Benefits of NAHLN messaging

The NAHLN messaging system is designed to provide benefits to each of its participant groups. The member laboratories will have a single mechanism by which they can



submit laboratory results for all participating Veterinary Services programs. This will greatly reduce the amount of double data entry into program-specific web-forms, spreadsheets, etc. In turn, the programs themselves will get a single source for all relevant test results in a common, standardized data structure with appropriate identifiers for joining back to program-specific databases. This will ensure consistent data content and representation, making analysis at a national level much more informative. And animal health authorities at both state and federal levels will have a single source for laboratory data relating to their own jurisdiction. These benefits come at no small cost in terms of need to standardize far more precisely than would be required for a single-purpose manual data-entry system. The NAHLN messaging architecture provides for this standardization.

### **System Architecture**

The NAHLN messaging system is based on a hub and spokes model. Each node in the network communicates directly with only one other node, the hub. The hub functions as a message broker to route the information to any and all appropriate end points. In initial implementations the hub (the Laboratory Messaging System or LMS) is one and the same location as the central NAHLN database.

All aspects of the NAHLN messaging system are designed to create data interoperability through enforcement of open standards rather than by requiring any proprietary implementation hardware or software platforms. Where specific software, tools, or platforms are mentioned, they are for example purposes only. Any system capable of connecting to the Internet can be programmed to participate in NAHLN messaging. Some tools make this programming easier, but they are not a requirement.

NAHLN messaging is by design a tool for sharing information between organizations. The requirements for what information must be shared in order to participate in the NAHLN have been established by the NAHLN Coordinating Council and surveillance program designers, and are beyond the scope of this document. The mechanisms for implementing this information sharing have been formalized in the NAHLN messaging standards as implemented under the guidance of the NAHLN Information Technology (IT) Committee. These standards fall into three broad categories: Messaging Standards include the structures and protocols for assembling the required information and routing instructions into standard format for transmission over the NAHLN messaging system. Data Standards include coding systems and other data format rules to ensure interoperability of message content. Security Standards include rules for authenticating senders and receivers of data, ensuring the integrity and confidentiality of data as they travel over the Internet, and enforcing access control rules established for recipients of messages as well as direct access to the data in the NAHLN database.

### **Messaging Standards**

The NAHLN uses Health Level Seven<sup>®</sup> (HL7<sup>®</sup>)<sup>2</sup> standards for its message structures and interaction protocols. This standard is widely accepted internationally as the standard for

---

<sup>2</sup> Health Level Seven and HL7 are the registered trademarks of Health Level Seven International.

human medical and veterinary messaging. Other programs using HL7 include the Centers for Disease Control and Prevention (CDC) for all of its Public Health Information Network (PHIN) and Laboratory Response Network (LRN), the Food and Drug Administration (FDA) for the Food Emergency Response Network (FERN), and the Department of Health and Human Services (HHS) for the National Health Information Infrastructure.

A complete understanding of HL7 is not required in order to successfully construct and transmit NAHLN messages. Using this document, the Implementation Guide(s), and other supporting documents, you can work your way through the messages used. You can be confident that the bits of HL7 that you pick up along the way will be applicable to just about any other veterinary messaging project you may take on in the future. The more understanding of HL7 you can pick up, the more some of the design decisions that have been made will make sense to you.

### **Data Standards**

Data standards, code lists, etc., tend to be maintained by domain experts in various medical specialties. As a result, different standards are used in different parts of the NAHLN messages. Each standard provides solutions for a more-or-less specific application in messaging and data systems. One standard is used to encode test types, another for units of measure, and another for names of species, diseases, pathogens, etc. Some small lists are maintained by HL7 or even by the NAHLN IT Committee itself. This document and the terminology service will guide you through the process of determining which system and which subset of codes is needed in each application.

General medical and veterinary nomenclature is maintained in the Standardized Nomenclature of Medicine (SNOMED). This standard, besides providing coded concept identifiers and synonyms for over three hundred thousand concepts, provides these concepts in a hierarchical context that allows generalization on various axes. Thus you can know from the relationships assigned to the concept identifier for Classical Swine Fever (28044006) that this is a disease due to Pestivirus (72730002) which is a virus (34014006). You can also quickly find related diseases and so on.

Laboratory and clinical observations are identified using the Logical Observation Names and Codes (LOINC®). These codes are actively maintained and updated as new types of tests are developed. The maintainers at Regenstrief Institute in Indianapolis can and have added codes for tests used only as part of the NAHLN.

Other data standards include the use of standard abbreviations for units of measure, standard formats for dates and times, and standard representations for object identifiers, addresses, phone numbers, and so on. These are spelled out both in this document and in the Implementation Guide(s).

## Security Standards

Computer security professionals like to talk about “CIA security.” CIA, in this case, stands for Confidentiality, Integrity, and Availability. The NAHLN network includes features designed to ensure that each of the areas of CIA security are addressed.

Data that flow through the NAHLN network are potentially highly sensitive. They cannot be kept private—completely unshared—but must have their distribution carefully controlled. The principle of data confidentiality says that access to the data is restricted to those who “should” have access to them. The decision of who should have such access is a policy decision that takes place at a higher level. The NAHLN messaging system enforces those decisions using role-based access. Each user or system has one or more specific roles assigned. Access to read or write any given piece of data is based on these roles. Identifying and authenticating—proving the identity of—users accessing the NAHLN system involves another set of technical controls.

### Technical Security Details

Encryption of data flowing over the network prevents wire-tap type reading of the data. User IDs, passwords, and public key certificates are used to verify the identity of systems requesting access. These services are provided by an Internet Engineering Task Force (IETF) standard technology known as Transport Layer Security (TLS), also known as Secure Sockets Layer (SSL). Many people are familiar with this technology from the “little lock” on their web browsers.

Integrity is one aspect of data quality. Data entrusted to an information system must not be changed in content or meaning by the system itself. The NAHLN has other standards such as standardized methods and proficiency testing to ensure the correctness of results produced by NAHLN labs. Once these results are in the system, they must remain unchanged by system errors, malicious tampering, etc. If they are found to have been entered in error, the required changes must be documented to ensure that the original values are noted as errors and the corrected values are used in analysis. Various technical features of the NAHLN network serve to enforce data integrity. Many of the coded values have mathematical checksums to detect typing or transmission errors. All coded and many numerical results have sets of allowable values in order to catch incorrect (impossible) values.

Availability means that when animal health officials or others with legitimate reasons to use the NAHLN data need to access it, they can. Not all systems need the same level of availability. Some systems such as those that directly control spacecraft or medical devices must essentially never go off-line. Others such as retail systems or credit card authorization systems can tolerate a few minutes of down-time now and then. Discussion of availability often involves a number of nines. A system that is available 99.999% of the time is said to have “five nines” of availability. Each nine comes with a significant increase in purchase and maintenance cost. It is important that system designers specify enough, but not too many nines in their requirements. The NAHLN messaging system is designed to have high, but not spacecraft-level availability. It can afford to go off-line for an hour or two now and then for maintenance, etc. It cannot expect to be off-line for

much more than that—especially during an outbreak—without causing significant problems.

### **Requirements for participation**

In theory a lab could enter the data needed for the NAHLN message manually for each message. In practice, however, it is impossible to handle the kinds of volume for which the NAHLN was created without automation. Computer systems can automate extracting data from the LIMS, and translating and formatting those data into messages ready to send. Some LIMS systems make this process easier than others. This section looks at some of the features that facilitate messaging. Even more far-fetched is the possibility that a laboratory could send the NAHLN messages via some other transport mechanism than the Internet. In reality, there are some network capability requirements.

Perhaps most critical are professional staff members that understand the local terminology and help develop the translation maps needed to send data in standardized vocabulary. It is virtually impossible for a laboratory without such staff to be successful. Help will be available, but a complete outsider would not have the understanding of each lab's local usage necessary to perform the mapping without local assistance. This section outlines what you need, besides your towel, before embarking on a NAHLN messaging project.

### ***Laboratory Information System Capability***

It may seem overly obvious, but before one can automate messaging from—and to—the LIMS, programmers must have some means of accessing the data from the LIMS. This access may be direct, such as an ODBC (Open Database Connectivity) connection to the LIMS's internal database. More likely this will be access to a separate area of the database or another database that is populated by the LIMS on some schedule or in response to a trigger.

Whether or not the LIMS creates the accessible data in response to a trigger event, there must be some mechanism for detecting the event that requires a message to be sent. If the LIMS lacks the internal logic to send a specific trigger event, the messaging component will need rules for what value changes represent a state transition that should result in the creation of a message.

At least two commercially available LIMS systems can create and send a NAHLN Laboratory Result Message themselves. That doesn't mean the lab doesn't have work to do. All of the content mapping from local usage to the standard codes still has to take place. In these cases that is generally done in the LIMS's configuration settings somewhere.

Computers are very powerful, but very simplistic. While a human may be able to look at a case summary and just know that the serology tests came from the serum specimens and the histopathology results come from tissue specimens, the computer needs to be able to look up specific identifiers for each real world object they describe. Chances are the LIMS has identifiers that it uses internally. But chances are also good that it doesn't

show these to the users in the user interface. Understanding how the LIMS handles these identifiers will be an essential step early in the mapping process. Some examples of objects that must have identifiers are accessions, samples, orders, and results; plus, premises and animal identifiers as applicable.

Just as objects must have consistent, machine-readable identifiers, the descriptions of objects such as specimens must use some form of controlled vocabulary. Specimens may be anything from an entire dead animal to a tissue sample from a specific, defined part of a specific organ. Within the lab the details included in these coded values depend on the needs of the laboratory for its own production processes. For messaging these need to be translated to a standard set of specimen types. This is one of the many kinds of terminology mapping discussed in the Terminology section.

Test codes raise similar issues. Most if not all LIMS will have a controlled list of test types. These test codes often represent test catalog entries rather than specific results codes. Both order and result codes are important for communicating what was done and what was found during testing. Order codes clearly indicate the test requested. This test may actually be made up of multiple laboratory procedures resulting in more than one result value. Each result value has a code that clearly identifies the precise nature of the individual observation. In some cases the LIMS will not have results coded in the specificity needed for NAHLN messaging. In these cases, multiple sources within the LIMS may be needed to allow code mapping.

Understanding the range of possible answers that any given test result observation can give is essential to starting a NAHLN messaging project. There is a constant—and probably essential—tension between diagnosticians who want the full scope of the English language to express the full nuance of their observations on one hand, and epidemiologists who need to be able to perform classification and aggregate analysis on the other. For purposes of NAHLN messaging, the epidemiologists are right. The whole point of the NAHLN is to allow for large-scale response to surveillance and control measures. Only aggregation of results achieves this goal.

Just because the results field or fields in a LIMS have a plain text datatype does not mean that all is lost. In many of these systems, the number of distinct values entered into these fields is controlled either by the user interface or by laboratory data entry procedures. A data analyst can perform some simple queries against these data to determine the scope of the problem. Eventually, each distinct value possible for the covered tests will have to be mapped to a standardized vocabulary of possible results. Any “new” values that appear are going to require system maintenance. It is obviously best that the system enforce a controlled list of possible results up front.

Ultimately the messages sent via the NAHLN messaging system will consist of these standardized values in an Extensible Markup Language (XML) format designated by HL7 and refined by the NAHLN information technology committee. If the LIMS has the ability to generate XML output as a custom report, this may save a lot of the mechanical work of building the message. If it understands HL7 schemas, things will be even easier.

But even if the LIMS provides nothing but a flat view of the required data, it will not be an overwhelming task to do the conversion to the right format. So, the preceding data content issues are much more important than the format of the data coming from the LIMS.

### ***Network Capability***

It is possible to send Internet protocol messages via smoke signals—it has been done by students at Oxford University in England—but it is not something that will support NAHLN production volumes. In reality, a reasonably reliable Internet connection is a requirement for NAHLN messaging. The size of the connection depends on the volume of testing your lab plans to be able to provide. Be sure to design around surge capacity, not routine activity level.

Internet connections can be thought of as either outgoing only or as incoming and outgoing. An incoming connection is not absolutely necessary for NAHLN result messaging participation but increases the options for handling future Laboratory Order messages. Typical home and small business connections—digital subscriber line (DSL) or cable modem—may not provide a fixed IP address, making incoming connections hard to set up. Corporate and University networks are often protected by firewalls that prevent incoming connections to the machines inside the network without special configuration. Because there are security concerns with any incoming network connection, you may need to work with your network administrator to set up a special machine to handle incoming messages and forward them to the LIMS or messaging machine.

### ***Standard Terminology Support***

The NAHLN can and does provide help with the standards side of the translation but only subject matter experts in each member lab can determine what each term or code in the LIMS really means as used in that laboratory. The effort to standardize terminology and coding is not intended to force subject matter experts to use the NAHLN vocabulary. Its goal is to define what the subject matter expert is saying precisely enough that it can be translated into computable codes. This process often involves a significant conversation between a subject matter expert and an informaticist in order to come to a single, unambiguous meaning for each word, phrase, or code used in the source system.

When development of standard terminology mapping seems to be impossible, one of two things usually apply. In some cases, the standardized terminology system may lack a concept that matches the word being mapped. In these cases, the concept can be created either through a process called post-coordination or by modeling a new concept in the standard. For the current version of the NAHLN, post-coordination is not supported, while new concept creation is handled by the centralized terminology service. This service will require a strong supporting argument for why the new concept is needed, but a good mapping effort will have developed most of the needed material.

The second reason why mapping may fail is that the word or phrase in the LIMS is either ambiguous or doesn't actually define a single concept. At this point don't be surprised to hear the informaticist ask something like, "What exactly, are you trying to say?" The

answer may have to involve a significant bit of teaching virology, serology, etc., to the informaticist but in the course of that, the single, unambiguous concept intended will come out. This may then require minor changes to the source system if, for example, it turns out that the same word has been used to mean different things in different cases. It may be a matter of determining that the word has different unambiguous meanings in different contexts that can be determined from the system. This is then *just* a programming problem.

For this process to work, the laboratories must be willing to devote a reasonable amount of time and effort by one or more subject matter experts and one or more database/LIMS experts who may, but probably won't, be one in the same person. This can be a very fulfilling and educational assignment, but it does require commitment.

### **NAHLN System Account**

Not just *anybody* can send data to the NAHLN. Before your lab can send data, you will need a system account with a "File Upload User" role in the NAHLN/LMS system. As a NAHLN lab, you probably need several other roles as well. These roles are for real people rather than systems and require those individuals to obtain a USDA eAuthentication account. But the upload *system* is the one that can send results messages.

To obtain an account for messaging contact NAHLN at [NAHLN-IT@usda.gov](mailto:NAHLN-IT@usda.gov). Currently, you will need to fill out a 513 form requesting the account. The Enterprise Messaging Service uses token-based authentication and when your account is established NAHLN will provide the token. The account thus established is a 'system' account, so while a contact is needed, the account itself is specifically for messaging as in the past with the accounts used for Rhapsody. Should personnel in a lab need access to other tools, such as the Enterprise Management Tool (EMT - not currently available), individual accounts will be established. When the EMT is available NAHLN will provide account instructions; access to the tool is managed through the User Management System (UMS).

To submit results your laboratory must also be registered in the NAHLN Laboratory Registry. The information must include the lab's Premises Identifier Number (PIN) or Non-producer Participant Number (NPN) from ADT. If you need assistance obtaining a PIN or NPN please contact NAHLN or your state's state veterinarian.

### **What to send and when to send it?**

Of course, before one can send meaningful NAHLN messages one must have something to send and a reason to send it. Conversely, it wouldn't make sense to send everything a lab does to the NAHLN messaging system. The line between the two will undoubtedly shift as more and more tests come on-line in the NAHLN, but initially messaging is limited to a small number of tests with significant national importance.

Unlike some event reporting systems that only need to receive reports of positive findings, the NAHLN needs to receive results from all tests performed under the NAHLN protocols. Currently, only final results of NAHLN tests are sent via NAHLN message.

Partial *accessions* can and should be messaged, especially when some tests take significantly longer than others to complete, but preliminary (unverified) results of individual tests are not messaged.

Note that one special case is for foreign animal disease investigation testing “presumptive positive”—non-negative test results pending reference laboratory confirmation—are *not* to be messaged prior to confirmation. While this may not reflect a scientific risk assessment—these results *are* sent via email, a *much less* secure mode—it is USDA policy.

### ***Supported diseases and tests***

The NAHLN was originally created to support surge capacity in testing for eight foreign animal diseases of national significance.

- African Swine Fever (ASF)
- Avian Influenza (AI)
- Classical Swine Fever (CSF)
- Contagious Bovine Pleuropneumonia (CBPP)
- Exotic Newcastle Disease (APMV-1)
- Foot-and-Mouth Disease (FMD)
- Lumpy Skin Disease (LSD)
- Rinderpest (RP)

Later Mother Nature stepped in, as she is wont to do and rearranged the priorities. The first imported case of bovine spongiform encephalopathy (BSE) raised the priority of BSE testing, and the NAHLN was enlisted to communicate results of the enhanced surveillance program. At about the same time, chronic wasting disease (CWD) in cervids (deer and elk) and scrapie in sheep and goats were becoming growing concerns. Several years of vesicular stomatitis (VS) outbreaks in the western U.S. led to inclusion of that disease. This resulted in the next set of diseases being added to the list:

- Bovine Spongiform Encephalopathy (BSE)
- Chronic Wasting Disease (CWD)
- Scrapie
- Vesicular Stomatitis Virus (VSV)
- Spring Viremia of Carp
- Viral Hemorrhagic Septicemia (VHS)

Eventually even more routine testing has been added under the Business Process Improvement program:

- Pseudorabies (PRV)
- 

As NAHLN standard methods for these diseases are distributed to the laboratories, their results will be sent via the NAHLN messaging system.

A major force behind expanding the scope of tests messaged through the LMS was the rapid emergence of Porcine Epidemic Diarrhea (PED) and related Swine Enteric Coronavirus Diseases (SECD). This outbreak created the type of data overload that the



NAHLN was designed to handle. While it took significantly longer than it should to bring these tests onto the system, it eventually proved the value of seamless routing from labs to LMS to other data systems. In this case the eventual recipient was the Emergency Management and Response System (EMRS).

Currently, tests are added to the NAHLN/LMS result messaging system one at a time. Check with NAHLN to see which of these tests the system may be able to process. In the future, it is the goal of the VS CIO that *all* test results that must be reported to VS will be accepted via the NAHLN/LMS result messaging system. The systems to handle this generalized test result reporting are being developed. Then the only remaining issue will be the willingness of individual disease programs to accept their data in this format.

### ***Message transmission trigger***

The timing of NAHLN messaging is very important. The system is designed to support the rapid response to an introduced, emerging, or reemerging disease outbreak. Any delay introduced by the messaging system will detract from the overall effectiveness of the system as a whole. On the other hand, the national disease control systems are going to be acting on the basis of these results. Therefore, only finalized, verified, and validated results should be sent.

Designing the local triggers for NAHLN message transmission should take this balance into account. In general, as soon as a NAHLN test has produced final results, a chain of events should be initiated that will bring these results to the attention of the laboratory and or state animal health authority personnel responsible for verifying the final results. As soon as this verification is complete the results should be sent via the NAHLN messaging system. Depending on the approval process in each laboratory's jurisdiction, this final verification and validation may require approval by more than one person. Workflow systems should be designed to facilitate this approval process. Some commercially available LIMS provide a button to perform this release of results upon verification and validation.<sup>3</sup>

### **Tools of the Trade**

When you get serious about any endeavor, one of the first steps is to gather around you the necessary "tools of the trade." NAHLN messaging is no different. The three most important of these are this guide, the implementation guide, and the program-specific messaging guides. Another very valuable tool is the message mapping template, which assists in assigning the proper fields in your local database to the fields in the HL7 message. Always check for updated versions of these documents before creating or updating message structures. Another essential tool is a set of valid sample messages. Sources include Dr. Jim Case's hand crafted sample messages as well as fellow NAHLN laboratories that have already tested messages against the new structure. In addition to these NAHLN-provided tools, there are a few software tools that will help you

---

<sup>3</sup> Note that other uses of the NAHLN Messaging infrastructure may make different trade-offs in this area. For example USALIMS can be set to message to USAHERDS animal health software. Because these systems are more likely to be under single management timeliness was considered more important than waiting for finalized results. In this case, as soon as results are entered, they are sent.

considerably in understanding the message structure and creating and validating messages. Most of these are freely available. Others are commercial products that have proven to be useful.

### ***Validating XML Editor***

XML documents follow some specific rules about their structure and content. The most important are well-formedness and validity. The structure of XML consists of tags that enclose content. Each begin tag must have a matching end tag, and they must all nest in a logical order. This level of structure defines what is called being well-formed. This is a trait of the XML data file by itself. Another level of rules is defined by another XML document known as a “schema.” The schema defines specifically what tags are expected to appear in what order, and in many cases even what the content of those tags may be. An XML document that follows all the rules specified in its schema is said to be “valid.” Validity is a trait of an XML data file with reference to its schema. You need both in order to check the data file for validity.

XML files, both the data files and the schemas themselves are simple text files. As such, they are viewable and editable using any text editor. Anyone up for editing a NAHLN message using “copy con” and “edlin?”<sup>4</sup> However, it is much easier to work with these files using an editor that provides features such as syntax highlighting, auto-completion, pretty printing, etc. A popular tool for this application is XML Spy. XML Spy is a commercial tool available for trial download at: <http://www.altova.com/xmlspy.html>. Another option is oXygen, a Java-based multi-function XML editor that will run on most platforms such as Mac or Linux as well as Windows. It is available at <http://www.oxygenxml.com>. Both of these editors are commercial products (free for evaluation, but require payment for ongoing use.) There are freeware XML editors out there with a wide variety of features. The one essential feature is that they provide XML schema validation. These tend to come and go, so searching for “validating xml editor” is probably the fastest way to find something that works for you.

Many programming editors and other tools provide these features as well. My current favorite in the Windows world is Notepad++ <http://notepad-plus-plus.org/>. There is a very good plugin, XMLTools, to do schema validation. This plugin will also format your XML by adding line breaks and indents to make it more human readable. XMLTools can be installed from the Plugin Manager plugin once Notepad++ has been installed. Both Notepad++ and XMLTools are free.

If you happen to be doing this work on a Mac or Linux you already have a tool called “xmllint.” It is part of the libxml2 library. Typical of Unix command-line tools it has a huge number of options for validation, formatting, etc. And typical of Mac and Linux, it is relatively easily integrated into editors: both Unix-style like vi and emacs and GUI-style such as TextWrangler.

---

<sup>4</sup> Not old enough to get these references? In the very early days we didn't even have full file text editors, just one line at a time. These were a couple of the very basic ways to enter and edit text files.

Even if you can't or don't want to install an editor, you are going to need a tool to check the well-formedness and validity of your test messages. There are a number of on-line schema validators available. Examples include: <http://www.utilities-online.info/xsdvalidation/>, <http://www.corefiling.com/opensource/schemaValidate.html>, and <http://www.freeformatter.com/xml-validator-xsd.html>. As the name implies, the later also offers a formatter to make your XML easier to read if the tool that creates it does not auto-indent, etc. The core filing tool offers similar formatting.

“But” you might ask, “I’m not going to be doing any hand-editing of my NAHLN messages. Why would I need a fancy XML editor?” The truth is that once your messaging system is perfected, you probably won’t use it very much until the next major change to the system. But, while you are working the kinks out of your messaging, the editor can save a ton of heartburn as a debugging tool. You want to start by having the messages saved to a file instead of being sent to the NAHLN (even the test server). These are text files that you can read. If you read them in a validating editor, you can tell the editor to associate a schema with the message. By downloading the latest schema from terminology website *Resources* section and using it to validate against, the editor will show you all your errors.<sup>5</sup> The syntax for referencing a schema is a little complicated, but a good editor will insert that for you via a menu item or tool-button. The better the editor, the more user-friendly this validation process is. Most commercial editors at least allow you to jump right to the point where validation failed. The free options at least give a line number where they validation failed. This will most often be because an expected element is missing immediately before the point of failure. But the real magic is that you can try hand-correcting the errors and revalidating. This is a good way of understanding why the schema is unhappy with your message.

Most mapping engines or interface engines will provide some sort of XML editor and or validator. If you are using one of these tools, be sure to check out its features.

There are also programming libraries and standalone tools for validating XML. The best known of these is Xerces, an open source project that is part of the Apache universe. The Java library can be downloaded at: <http://xerces.apache.org/xerces2-j/>. They also provide C++ and PERL versions. Of course, you have to write your own code to use these libraries.

XML experts doing very complicated things often find problems with any given XML Validator, but it is wise to always assume that the tool is right and your work is wrong until you can prove otherwise. Most of the NAHLN work is not so complicated as to raise issues with the tools’ validation.

### **Schema Files**

To work with XML validation you must, of course, have or be able to point to the latest schemas. There is a base schema that provides the rules that apply to *all* NAHLN Result

---

<sup>5</sup> To be precise, it will find all the XML validation errors. Once you have worked through the message structure and have an XML file that validates, you will be almost ready to send messages. There are still content issues to work through and they will be covered in the chapter on *How Do You Know Your Message is “Correct?”*

Messages. At one time there was an effort to develop program-specific schemas derived from the base schema, thus the word “Base” in the schema file name. Now the “Base” schema is used for all messaging.

Copies of the schemas with the file names reflecting the version number are available for download in the Resources section of the Terminology website. These are posted by the authors as they are updated and validated. The moderator keeps the latest versions of most documents in the *Resources* section.

You can also link directly to the most current version of each schema on the *Resources* tab at the terminology services web site. At that location the filenames lack the version information. The reason for this is to allow you to point your XML tool to one location and have that stay current even as new versions are released. There is a URL for the base schema without the version number extension. For example the current version (NAHLNResultBasev2.1.xsd as of this writing) of the base schema can be found at <https://vtsl.vetmed.vt.edu/nahln/Documents/NAHLNMessageSchemas/NAHLNResultBase.xsd>. When this schema is updated that link will still point to the latest version.

If you are downloading schemas to a hard drive, you probably want the version number to help keep them straight. If you are just including a URL and reading directly from the web, it may be more convenient to use the generic versions and let the web site worry about keeping it current.

The rest of this chapter will assume that you have a way to look at the message structure and schemas and to check validity of files you create.

## NAHLN Result Message description

This *Hitchhiker's Guide* is not intended to replace the *NAHLN HL7 Implementation Guide For Laboratory Results Messaging*, but to provide a how-to guide to supplement the technical detail in that guide. For a complete description of the message structure, please refer to the *Implementation Guide*. This section will give you a framework in which to apply the details supplied in the *Implementation Guide*. The NAHLN result message is based on a specific set of constraints on a fully specified message. What this means is that the NAHLN IT Committee has picked the parts of an existing message structure that it wants to support and ignores the rest. This restriction is known in HL7 parlance as a “message profile”.

### **Abstract Message Definition**

HL7 messages, in their original format, consist of a series of lines of text known as “segments.” Each segment is made up of fields that can in turn be made up of components. A message structure is defined by the order in which segments can or must appear in the message and if and when any of these may repeat. To make it easy to see a message's structure at a glance, message structures can be displayed in a table format known as Abstract Message Definition. In this format, each segment or group of segments is shown as a line. Optionality is shown by bracketing the optional portion in

square brackets ( [ ] ). Repeatability is shown by bracketing the repeatable part in curly brackets ( { } ). The abstract syntax definition for the NAHLN results message is:

### NAHLN Constrained OPU\_R25 Abstract Message Structure

| <u>OPU^R25^OPU R25</u> | <u>Unsolicited Population/Location-Based<br/>Observation Message</u> | <u>HL7<br/>Chapter</u> |
|------------------------|--|------------------------|
| MSH                    | Message Header   | 2                      |
| PV1                    | Patient Visit  | 3                      |
| OBX                    | Observation on Visit   | 7                      |
| [{ROL}]                | Role   | 15                     |
| {                      | -- ACCESSION_DETAIL begin  |                        |
| {NK1}                  | Next of Kin  | 3                      |
| [                      | -- PATIENT begin   |                        |
| {PID                   | Patient  | 3                      |
| [{                     | -- PATIENT_OBSERVATION begin   |                        |
| OBX                    | Observations on Patient  | 7                      |
| }]                     | -- PATIENT_OBSERVATION end   |                        |
| ]                      | -- PATIENT end   |                        |
| {                      | -- SPECIMEN begin  |                        |
| SPM                    | Specimen   | 7                      |
| [{                     | -- SPECIMEN_OBSERVATION begin  |                        |
| OBX                    | Observations on Specimen   |                        |
| }]                     | -- SPECIMEN_OBSERVATION end  |                        |
| {                      | -- ORDER begin   |                        |
| OBR                    | Observation Order  | 4                      |
| ORC                    | Common Order   | 4                      |
| {                      | -- RESULT begin  |                        |
| OBX                    | Observation Result   | 7                      |
| }                      | -- RESULT end  |                        |
| }                      | -- ORDER end   |                        |
| }                      | -- SPECIMEN end  |                        |
| }                      | -- ACCESSION_DETAIL end  |                        |

From the top line of this table you can tell that the name of the message is, “Unsolicited Population/Location-Based Observation Message” and that its structure is abbreviated as, “OPU\_R25.” The rest of this chapter will walk through this structure and explain in general terms what each segment or group is for. We will not try to take the place of the implementation guide, but rather to give broad, general context.

### **Observation Population Unsolicited (OPU)**

So, what's up with that complicated message structure name anyway? The NAHLN asked me to send this information. So, “how is it unsolicited?” and other mysteries of the universe will be answered—we hope—in this section.

Regulatory and disease-control laboratory testing is very different from individual case diagnostic testing. For one thing the individual patient is not the center of attention. Rather the population—herd or flock—or location such as a specific farm or surveillance area is of central importance. The results of these tests tend to be aggregated at the lab level into what are commonly known as “accessions” but may go by other names. They

also relate to investigations or “cases” at the state or federal level rather than to individual practitioner medical records. These factors resulted in the creation of a message structure that, while similar to the patient specimen-based laboratory results message, can be seen as, “inside out.” The accession becomes the container and may contain a number of patients.

These results are *unsolicited* not because no one asked for them, but because the trigger event that causes them to be sent occurs in the laboratory. The event is simply that the results are complete and finalized and ready to send. Compare this to the case where the lab might complete all the results but not send them until another system requested results through an HL7 query message. That would be a *solicited* results message. Even if the lab has an order message to which it is responding with this result message, because the trigger event was test completion rather than a query, this is an *unsolicited* message. Be careful when listening to casual conversation about the NAHLN. Some less HL7 immersed people will use the term *solicited* to mean in response to an order message. This is *not* correct HL7 usage of the term, but it is out there.

Another question you might ask is, “What do you mean by Location-Based Observation?” This involves one of the more unique aspects of veterinary laboratories; that is, they test environmental specimens in addition to animal specimens. If you have the opportunity to study other HL7 messages, you will notice a difference in the messages used in the NAHLN and most other implementations of laboratory messaging. The *patient* is optional. This sometimes causes grief for users of patient-centric information systems, since all testing is based on something from the patient. In environmental testing, however, there is no patient, only the place (location) where the samples came from. Because the patient information is optional in this message, there will be some rather irritating features of the message that will be discussed later on. Suffice to say that some of the limitations imposed by the HL7 structures result in strange message instances. Most of these limitations simply result in information being repeated in ways that don't seem important to humans. But to automated systems like Marvin the Manic Depressed Robot, dealing with this kind of redundant material is hardly enough to distract from the annoying buzzing in their diodes. So, we let the computer worry about that.<sup>6</sup>

So, these details result in a unique message, trigger event pair and its corresponding message structure, “Unsolicited Population/Location-Based Laboratory Observation Message.” Because that is a lot to include in message headers, and a mouthful to say when referring to the message, it has an abbreviation of “OPU\_R25.” The OPU part represents an abbreviation for the message structure name, while the R25 part represents the unique trigger event (i.e. results ready to transfer). Since the same message structure can be used for different purposes, the trigger event that resulted in the message being sent needs to be included when the message is sent.

---

<sup>6</sup> Seriously, it is very hard to “use up” the processing power of a modern computer doing this kind of work. It is much better to “waste” a little computer processing than to spend excessive effort eliminating these minor redundancies. And because XML data are simple text “files,” they are *much* smaller than similar data in binary formats such as Excel, even when the messages look large to human eyes.

With all that behind us, let's dive into the message structure OPU.

### **Message Header Information**

You can think of the Message Header (MSH) as an envelope in which the interesting part of the message is contained. This header includes a destination address, a return address, a postmark time stamp, and a little additional housekeeping information.

Other than the message control ID, processing ID, and the system date/time for the message date/time, this segment can be almost completely hard coded. For systems other than the NAHLN/LMS, these address elements vary and are very important.

### **Addresses:**

Four fields in the MSH hold the source and destination addresses. The sender and receiver are each identified by application and facility. The facility is required for each and the application is required if the facility has more than one application that may send or receive messages.

### **Message Control ID:**

One extremely important piece of this housekeeping information is the Message Control ID. It is essential that the NAHLN system be able to identify a specific message instance in the system. The Message Control ID serves this purpose. Each sending system must be set up to create a unique identifier for each and every message it sends. These *may* end up duplicating the ID's created by other sending systems and will be distinguished by the sending facility and/or sending application identifiers. Sometimes our messages go wrong. This is covered in detail later in this manual. In those cases the same result must be resent in a new message. The new message needs a new message control ID. For this reason, the message control ID is not equivalent to a result ID. (The latest version of the schema does include a unique result ID in a different location.)

### **Processing ID:**

One additional bit of information can be very useful while developing NAHLN message. The MSH.11 field, "Processing ID" is used to differentiate "real" production messages from those used for testing and debugging. As a lab starts developing its messaging capacity, it may need to actually send messages in order to test transmission, etc. These messages are meant to be entirely ignored by the receiving end other than basic message validation needed to return an acknowledgment. These messages are said to be in "debug" mode designated by a "D" In the NAHLN, the "test" mode, designated by "T" is used to identify messages being sent to test the lab's capacity to message for a given test or program. These messages are meant to be evaluated by NAHLN program staff prior to approval of production messaging.

### **Message Profile Identifier:**

As the NAHLN expands its scope in the number of diseases for which it requires laboratory data, there will be different kinds of data required for different programs. This has led the NAHLN development team to adopt the concept of different message profiles for each disease or set of diseases where the amount and kinds of data differ. Each

profile is a proper constraint of the full OPU result message structure. Currently, the only variation in NAHLN profiles is the version of the “Base” schema. The message profile ID tells the NAHLN which version of the schema to validate against. For this to work the ID must be the name of the schema, including the version, but without the “.xsd” extension is used as the profile identifier.

### ***Visit (Accession) Information***

The message header is always followed by a “Patient Visit” (PV1) segment. The “visit” here is a laboratory accession (or “laboratory case,” or any other term describing a group of related orders within the lab). This segment may be used in either of two general approaches. In one strategy, when all samples from an accession are finalized at approximately the same time, they may be combined into a single message with one patient visit followed by any number of specimens, orders and results. In this case, the single patient visit segment provides many of the details common to all specimens and tests. In another approach, each result may be sent in its own message. In this case each message will contain a patient visit segment but each of these will contain the same information pointing to the single accession from which the samples came. Either approach allows the NAHLN to recombine the results for analysis and or further inquiry to the lab.

The most important information communicated in the patient visit segment is the accession number. This uniquely identifies the case to the laboratory. This segment also contains the laboratory diagnostician (case coordinator) with primary responsibility for the case. For detailed usage see the Implementation Guide.

Additional information about the “visit” can be communicated in observations about the visit. This is done in HL7 by following the patient visit segment with one or more “Observation” (OBX) segments.

In the first two iterations of the NAHLN Messaging, the reason for submission was sent as an observation on the visit. This allowed one code to apply to the whole accession the way most labs do this. However, the corresponding Order message has no visit; there is no accession yet! For that reason, this item has been moved to the individual test orders. It repeats and will essentially always be the same for all orders in an accession. As mentioned earlier, the cost in computer processing is a small price to pay for having the same information in the same place in both messages. Other observations on the visit may be included as appropriate in the future.



## ***Related Parties Information***

### **(Submitter, Premises, Collector, Other interested parties)**

While most regulatory and disease control testing does not follow strict legal chain of custody rules, it is nevertheless important to keep track of individuals and other parties that participate in the testing. These entities are said to be playing a “role” in the testing. Some important roles include the specimen collector, the submitter, the attending veterinarian, and even the premises being tested. This information is communicated in a series of “Role” segments that follow the patient visit segment and any associated observations. As with most things in the NAHLN, the Role segment is designed to unambiguously communicate the nature of the role as well as the identity of the entity filling that role. So, for all Roles we include a code that specifically represents the role being played by the entity represented in the segment data. For people we also get one or sometimes two specific coded identifiers. For premises, the Animal Disease Traceability (ADT) Premises Identifier is essential.<sup>7</sup> If no PIN is available, an absolute minimum requirement for the PREM Role is the state of origin.

#### **One Premises?**

Is “premises” a singular or plural? My word processor wants to make it “premise.” What's going on with this word?

“Premises” as used in the ADT is a legal term meaning a parcel of land and all the buildings on it. It derives from the word “premise” meaning “a proposition supporting or helping to support a conclusion.” The definition of the premises is listed first in a legal paper and thereafter referred to as “these premises.”

So, “premises” is a plural word even when used to describe a single parcel of land. The word is the same when referring to multiple “premises” as when referring to a single “premises.” We should probably still say, “these premises” the way the lawyers do, but we don't usually.

By the way, Microsoft Word's grammar checker hates this sidebar, but it is correct; really.

For detailed usage of the Role segment, see the *Implementation Guide*. Uses of some of these roles are easy to confuse with other entities identified throughout the message. These confusing uses are discussed further in the section on “Keeping Similar Concepts Straight.”

## ***Accession Detail***

Each of the items addressed so far relates to an entire message and appears once at the beginning of the message. There may be multiple observations or roles, but there is one of each type that applies to the whole message. Following this, however, we begin to get into repeating groups. They don't *have to* repeat, but they can. An accession detail is all information relating to a single patient. Given that this message structure was designed to handle testing that may not be on a specific animal or even species; what do we mean by this? We are using the word *patient* to mean a single unit of epidemiologic or therapeutic significance. This is consistent with the latest version of AVMA's Model Veterinary Practice Act section 2 definition of a patient, “Patient' means an animal or

---

<sup>7</sup> Under the current concept of the national Animal Disease Traceability system a state animal health authority may assign either Premises Identification Numbers (PIN) generated centrally by VS or state generated Location Identification Numbers (LID). Here we will use Premises Identifier or PIN to refer to either type of ID.

group of animals examined or treated by a veterinarian.” So, if we are testing a herd of cattle and will treat or cull those that test positive, we need to treat each animal as a *patient*. On the other hand, if we are testing a flock of chickens, we will respond to any positives by treating or depopulating the entire flock as a unit. If we are testing a group made up of multiple species, each species constitutes a separate *patient* even if they otherwise would be grouped. Each *patient* in this sense has its own Accession Detail group. There can be as many Accession Detail groups as needed to describe the accession.

### **Animal Owner Information**

Each *patient* has by convention a single *owner*. We are using a lot of italics here, because we are using these words in slightly unfamiliar ways. Just as *patient* is any unit of significance that is treated as a unit, the *owner* may be the owner in the usual sense but is more correctly thought of as the person with primary responsibility for the *patient* animal, flock, or location. Each *patient* in this sense has exactly one *owner* while each *owner* may have one or more *patients*. Just in case you aren't confused yet, the *owner* information is sent in a segment called, “Next of Kin” (NK1). This is not a bow to the animal rights folks, but rather a legacy from the human health origins of HL7. In healthcare, the “next of kin” is the person most responsible for the patient in the case the patient can't answer for himself. This is the sense in which we use *owner* or, “next of kin.”

Next of Kin includes the relationship, name, identifier, and contact information for the owner or other responsible person. This segment occurs as the first segment of every Accession Detail. If one owner has multiple animals, flocks, or other *patients*, this segment is duplicated for each Accession Detail (this is the irritating part that we delegated to Marvin the Manic Depressed Robot earlier). For the most part, the message structure is designed to minimize this type of duplication. This is an exception. But just like Marvin, computers are very patient about this kind of monotonous duplication, so they probably won't complain.

To further complicate things, sometimes the laboratory may not know anything about the *owner* but still has to send the NK1 segment. In these cases it may send a virtually empty NK1 segment. Its only element is one required counter variable.

### **HL7 Legacy Issues**

HL7 is over thirty years old! In the bad old days, we had to save every possible character sent in a message. The original “pipe-delimited” (ER7) format for messages did not use names for anything except segments (those three letter abbreviations we keep using). The rest was inferred from position delimited by | (pipe) characters.

Keeping track of groups in that format was tricky. Parsers had to have rules for what segments they could see next and had to know from the next segment where they were in the structure. To make this work the first segment of any group needed to be required.

The patient (PID) segment in the OPU\_R25 message is not present in messages related to purely environmental testing. To conform to the general HL7 rule that first segments be required, each accession detail is “anchored” by the owner rather than having one owner with many patient groups.

## Animal Information

Not all testing relates to an animal or group of animals. This message can deal with testing empty premises for example. But if testing relates to an identifiable species, herd or flock, or individual animal, this information needs to be included in the message with one *patient* per accession detail. The information goes in a Patient Identification (PID) segment and related observations. (Some program-specific testing makes an exception to this rule because it *always* has the animal information on file from other sources and can link to it via the specimen identifier. But as a general rule, if there is an identifiable species, the PID segment should be sent with the animal information.)

Much discussion has gone into how much aggregation is appropriate when deciding what makes up the *patient*. This was discussed in the introduction to the section on the Accession Detail group. One way to see that the patient needs to be divided out from a group to a series of specific animals is that details in the Patient Identification section would be significantly different. If, as a made-up example, a chicken farm houses pullets in one house and adult breeders in another and tests birds from both groups, the age would be significantly different between groups but close enough to combine within groups. So, in that hypothetical case, you would want to send two accession details so that each of the two PID's could correctly describe the population of birds tested. You would not need to split out the two flocks into individual birds because they have common signalments and would be treated as a group for any necessary control measures.

Similarly, if tracking individual animal identification is important, you would need one Accession Detail for each animal. If they are combined, the Patient Identifier List will contain the identifier(s) for the group—ADT Group Identification Number (GIN) for example. This field can be confusing. Because it can repeat, it is tempting to create a group by listing the individual animal identifiers for the members of the group. That would be an error!<sup>8</sup> This is to allow listing alternate identifiers for the exact same entity. For example, during transition from breed registry numbers to ADT Animal

### Phantom Data

Throughout this Hitchhiker's Guide and even in the Implementation Guide you will occasionally read of elements that "must always be present" but that, when you look in the test program-specific guidance are allowed to be empty. How can this be?

One of the objectives of the NAHLN IT System is to minimize the overhead costs in getting lab results delivered to the central epidemiological databases. When it was found that much of the data that "must always be present" could be obtained from the order-entry system(s) by linking on identifiers carried in the message, the message designers realized that the system would still work even without these elements in the result messages themselves.

So in the end, these elements remain essential to proper functioning of the overall system, but they may in some cases come from sources other than the result messages. See the program-specific guidance for details.

---

<sup>8</sup> This concept and problem extend beyond the NAHLN and messaging. Many state veterinarians are facing issues with Certificates of Veterinary Inspection containing multiple identifiable animals squeezed into a single group record. Group records in those documents are also only appropriate for inseparable groups such as lots of nursery pigs. Individuals represented as a group do not process correctly in information systems without massive, time-consuming manual correction.

Identification Numbers (AIN) as primary on-farm identification of pure-bred animals, you might need to list both these identifiers for an animal.

Besides the identification for the animal or group, the Patient Identification segment contains most of the information you would collect for the signalment (demographics). This includes sex, date of birth, taxonomy, strain, and production class as well as information such as the date of death for necropsy cases. More information on how to designate each of these attributes is included later in the section on NAHLN Standard Terminology.

HL7 has a long history of very specific human hospital usage. In this context, the patient's last name became established as the minimum required bit of identifying information. As such, when the PID segment is sent, it must include something in patient last name (PID.5/XPN.1/FN.1). But most of the time our patients don't have a last name! Sometimes it isn't even one *patient* at all. So, what do we do about the last name field? We have to send the full element but populate with a phrase indicating that this isn't a real last name. In the NAHLN this has generally been, "Not Provided." In some zoonosis reporting systems, "Non-Human" is used instead to emphasize that the data are not from a human patient when they are received by a public health information system. Anyway, PID.5 in NAHLN messages will almost always look like this:

```
<PID.5>  
  <XPN.1>  
    <FN.1>Not Provided</FN.1>  
  </XPN.1>  
</PID.5>
```

If additional information about the animal or group needs to be included in the message, it can be sent as observations about the patient. This is in the form of possibly repeated Observation (OBX) segments. Some of the ways that this is used include, the descriptive age of the animal if birth date or approximation is not available, or associated clinical findings.

### Age Categories

Recording age rather than date of birth can be very problematic even when dealing with a single species like people. When dealing with various species and various disease control programs, the number of ways to classify age can explode. It is always preferable to record date of birth if possible. That said, it isn't always possible or practicable within program guidelines to do so.

When we must record age category of a patient we don't do so as structured data in the PID segment but instead as an observation we have made about the patient.

```

<OPU_R25.PATIENT>
  <PID>
  ...
  </PID>
  <OPU_R25.PATIENT_OBSERVATION>
    <OBX>
      <OBX.2>ST</OBX.2>
      <OBX.3>
        <CWE.1>46251-5</CWE.1>
        <CWE.2>Age Group</CWE.2>
        <CWE.3>LN</CWE.3>
      </OBX.3>
      <OBX.5>
        <CWE.1>28891000009105</CWE.1>
        <CWE.2>Neonatal Pig for Meat Production (organism)</CWE.2>
        <CWE.3>SCT</CWE.3>
      </OBX.5>
      <OBX.11>F</OBX.11>
    </OBX>
  </OPU_R25.PATIENT_OBSERVATION>
</OPU_R25.PATIENT>

```

Recent modeling of the concept of age group has used the “Animals in Context” ontology developed at Virginia Tech. Age grouping is always species and production type specific. We have not historically thought of it that way because in medicine and public health it is always assumed that we are speaking of *Homo sapiens* growing up and aging in a developed world society. For certain pandimensional beings a single game of Brockian Ultra Cricket may last longer than a typical human lifespan. So naturally age groups would be different for them. Likewise, we need distinct age group categories for each animal type. So, the codes we use to indicate age group actually come from the SNOMED organism hierarchy but from concepts that include development stage as part of their definition. These carefully selected codes can be found in the Terminology Services browser under the category, “age\_group”.

Animal taxonomy is probably the most critical field in PID. The default value for the *species* field is “human” so unless you are doing human testing this field is mandatory! By convention the species field holds the animal taxonomy with as much precision as possible within the coding system (SNOMED). This may be breed if an appropriate SNOMED code exists or it may only be genus if the species is not known.

## Specimen Information

The NAHLN Messaging system is all about sending results from laboratory testing on specimens of one kind or another. Thus, the one most consistent real-world *thing* to be associated with the results is the specimen itself. Especially for regulatory and foreign animal disease testing, it is vital that there be no confusion over which specimen(s) have produced which results. The Specimen group contains the information about the specimen as well as the following information about the tests performed on that specimen and the results from those tests.

The specimen segment itself provides us with unambiguous identification of the specimen and, if the specimen is derived from another via sub-sampling or pooling, the

identity of those source specimens. This segment also tells us what kind of specimen this is (such as serum, brain tissues, etc.) and if important, the anatomical location from which it was collected. Another critical item is the date and possibly time that the sample was collected. That is, after all, the most significant point in time if we are concerned about the actual condition in the animal(s). Finally, this segment gives some information on how the specimen is preserved and packaged, and its quality and condition on arrival.

The Placer Specimen ID (SPM.2.1) is used to send any identification that came for the specimen often designated by the submitter (placer) and attached to the specimen container. Ideally, SPM.2.1.3 can be populated with an OID to indicate the specimen identifier issuing authority. Otherwise, this subcomponent should not be populated.

Because many submissions come directly from industry veterinarians, rather than program specimen collection activities, they will not come in with globally unique specimen IDs (e.g. barcoded). Instead labs will need to report whatever ID did come on the submission. The placer ID's assigning authority may then be omitted or a text description "Farm ID" or similar if provided by the LIMS can go in SPM.2.1.2 just for human readability in knowing the type of ID.

The lab's Filler Specimen ID (SPM.2.2) is a critical bit of information. Because the lab issues this, it can ensure that it is unique. The ID which is often a database key combined with the lab's OID will be a globally unique identifier. When systems need to ask, "How many specimens were tested for X?" they can count unique values of this element.

As with the visit and patient, if we need to provide additional information about the specimen, we can do so via a series of observations about the specimen. These are not the results of any testing on the sample, even if the test is observational such as a necropsy. Rather, they are things that the lab may have observed about the specimen itself that might impact the interpretation of the test. Most such observations have coded values that can be sent as "quality" or "condition." That is the preferred method of sending this information. But more general observations about the specimen can go in observations about the specimen.

### ***Order Information***

Whether or not the order for testing has been transmitted electronically, the order information is part of the results message. The key pieces of this segment are the order numbers—both from the submitter and the lab—and what was ordered. This also provides the status of the order, which will always be final in the current form of the message, and the date when those results were finalized.

### ***Common Order Information***

For purposes of the NAHLN results message, the Common Order (ORC) segment functions as an extension of the Order. In other messages, these two segments nest in a slightly different way that creates an important distinction, but here they are more or less part of the same set of information.

Among the most important additional pieces of information is the Order Control that tells whether this is a new order, a status change, or a correction, among a lot of other options not used by the NAHLN. In the current result message implementation this is always either “NW” for new orders or “SC” for status change. When corrections or order/response pairs and partial order messages are implemented, additional options are likely to appear.

Also very important to the NAHLN is the Placer Group Number. This vaguely named field allows us to transmit the referral number for the Foreign Animal Disease (FAD) investigation or other case number used in the field. When a named Incident is created in Emergency Management and Response System (EMRS2) and testing is taking place as part of that incident, the EMRS2 Incident Name or identifier may be used as the Placer Group Number. In program testing without a referral number, the Placer Group Number consists solely of the program OID as the Assigning Authority. This uniquely identifies the program that ordered this test. That information, in turn, allows the NAHLN to reassemble all the data for a given investigation.

The rest of this segment is made up mostly of housekeeping fields described in detail in the implementation guide. These include the data enterer, verifier, the date/time the message was ready to send, and the status of the order.

### **Result Information**

In *The Hitchhiker's Guide to the Galaxy*, Arthur spends most of the story settling for something “almost, but not quite, entirely unlike tea.” It is only in the final chapter that he finally gets to visit *The Restaurant at the End of the Universe*. So, you see there is precedent for the meat coming at the very end. The NAHLN result message is no exception. Since you have been so patient in waiting to hear the results of your test, let's not keep you any longer from the actual results.

The Result group is nested deeply inside the Accession Detail, Specimen, and Order groups. Each of these groups can potentially contain many of the next layers. Similarly, the Order can have any number of results. For example, a foot and mouth disease serology might consist of specific tests for several different serotypes with the precise types determined by geography and other factors best determined by the specialist at the Foreign Animal Disease Diagnostic Laboratory (FADDL). Each individual value is considered a distinct result and gets its own Result group and Observation segment. Because we are not using human-only-readable notes segments, the Result group consists exclusively of the Observation segment.

This is another place where any ambiguity could potentially cause significant problems in interpretation of the message. The Observation (OBX) segment must make absolutely clear what observation is being reported and what the value is. For this reason, both the Observation Identifier (OBX.3) and the Observation Value (OBX.5) have tightly controlled code systems and value sets.



The Observation Identifier is always a Logical Observation Identifiers Names and Codes (LOINC) code. See the NAHLN Standard Terminology chapter below for more information on LOINC. To help in selection of exactly the correct LOINC code, the NAHLN has established a terminology service hosted at the Virginia-Maryland Regional College of Veterinary Medicine. This service is discussed in the terminology chapter.

Observation values can be any of several different data types depending on the type of test. Computers can get as crabby as Marvin the Manic-Depressed Robot if they get a data type different from what they are expecting. For this reason, the observation segment is explicit in providing the data type in OBX.2 before it gets to the value itself. See the later chapters on specific test types for detailed information on the different data types used to communicate different test results such as titers, isolates, and serotypes.

But what happens when we get a *really bad* specimen? In this case the lab doesn't test the specimen at all. In this case, we need to communicate that the specimen has been accounted for, but has not been, and more importantly will not be, tested. Again, we transmit the specimen details in SPM, indicating that the specimen is completely inappropriate (I) in SPM.23. But we have to send *something* in the result since this is a result message. In a pure HL7 world, we wouldn't send a result at all. After all, there is no actual result to send. But some testing programs expect a result for each specimen no matter what. So, to meet this requirement, we need to send a *result* that says, "We canceled this test." We do this by sending a result OBX with the result status set to "X". The rest of this OBX segment should reflect what *would* have been sent if the test could have been performed. The result value (OBX.5) and interpretation (OBX.8) are omitted or blank.

## NAHLN Order Message Description

In this chapter we will assume you have already familiarized yourself with the general structure of the NAHLN Result Message. Where the segments are the same, we have repeated much of the previous section here so that the section can stand on its own. Why the result comes before the order is something of a historical artifact. One would normally assume that you need an order before you can have a result. That is true, but in its beginnings the NAHLN had electronic results for orders that were submitted on paper. We hope that by the time you read this at least some orders will be electronic. If not, you are part of that beginning and we hope part of moving us on.<sup>9</sup> If you are not using order messages, you can safely jump to *Keeping Similar NAHLN messaging Concepts Straight* at this point.

A note is in order about the meaning of the word "order" as used here. "Order" may be the act of a clinician or regulatory veterinarian indicating that a test should be performed. It may be the entry of the information about the testing into a system such as USDA's Veterinary Services Laboratory System (VSLS) or Emergency Management and Response System (EMRS2). Finally, it may be the act of *ordering* a laboratory to perform the actual testing. (To keep sensitive computers happy, we use the more polite,

---

<sup>9</sup> Ten years after this section was first written we are still not actively messaging orders. Things change slowly in this business but once they kick in they move quickly.



“Observation Request” as the actual test order.) In the context of NAHLN Messaging it is the last meaning that applies. It happens that the Order Message contains much information that is derived from--or may be the source of--information in VSLs or EMRS. The Order Message always originates with some entity that is requesting a laboratory to perform some testing and directed to the laboratory to initiate an accession. Other uses of the information contained in the Order Message are secondary.

## Abstract Message Definition

The abstract syntax definition for the NAHLN order message is:

NAHLN Constrained OPU\_R25 Abstract Message Structure

| <u>OPL^O37^OPL O37</u> | <u>Population/Location Based</u><br><u>Laboratory Order Message</u> | <u>HL7 Chapter</u> |
|------------------------|---|--------------------|
| MSH                    | Message Header  | 2                  |
| {ROL}                  | Role Segment  | 15                 |
| {                      | -- <i>Order Group Begins</i>  |                    |
| NK1                    | Next of Kin Segment   | 3                  |
| [                      | -- <i>Patient Group Begins</i>                                      |                    |
| PID                    | Patient Identification  | 3                  |
| {[OBX]}                | Patient Observation   | 7                  |
| ]                      | -- <i>Patient End</i>   |                    |
| {                      | -- <i>Specimen Begin</i>  |                    |
| SPM                    | Specimen Segment  | 7                  |
| {[OBX]}                | Specimen Observation  | 7                  |
| {                      | -- <i>Observation Request Begin</i>                                 |                    |
| ORC                    | Common Order Segment  | 4                  |
| OBR                    | Observation Request Segment   | 4                  |
| {{                     | -- <i>Timing Begin</i>  |                    |
| TQ1                    | Timing/Quantity   | 4                  |
| }}                     | -- <i>Timing End</i>  |                    |
| }                      | -- <i>Observation Request End</i>                                   |                    |
| }                      | -- <i>Specimen End</i>  |                    |
| }                      | -- <i>Order End</i>   |                    |

The rest of this chapter will walk through this structure and explain in general terms what each segment or group is for. Where usage is the same as the result message we will reference the description there rather than repeating everything. We will not try to take the place of the implementation guide, but rather to give broad, general context.

## Message Header Information

The message header MSH segment is nearly identical to the MSH in the result message but with the “from” and “to” reversed. Now it is the ordering application that needs to generate a message control ID that will be used to keep track of the message sent, etc. The sending facility and application are those of the submitter, and the receiving facility and application are those of the lab.

## **Role Segments**

Just as in the result message, there are several specific roles in an order message. These must include a Submitter and may include additional repeats for other people. They may also include an ROL segment for the form of origin of the specimens formatted as it is in the result message.

An additional role type is needed for report recipient. Any report recipient roles should contain at least one form of contact information used to send the report. Most systems will prefer not to report based solely on this segment but will use the segment to look up an already registered party in their lab system and use the contact information there. The role instance id, in this case, may be the party identifier from the laboratory's system, if provided.

Most labs will require a ROL segment with type PAY that includes the lab-assigned billing account number in ROL.1. Without this information, the lab will have to contact the submitter to determine billing, which could delay testing.

## **ORDER Group**

Beware the naming of the groups. They sound similar to those in the result message but sometimes represent a different level of aggregation. Such is the case with the ORDER group. It is more like the `ACCESSION_DETAIL` in the result message. It is anchored by a Next of Kin (NK1) segment and repeats for each *patient*. Within the ORDER Specimen repeats and within that we get the actual `OBSERVATION_REQUEST` groups that would correspond to the ORDER group in the result message.

The Next of Kin (NK1 aka "Owner") segment and the PATIENT group are used exactly as they are in the result message. The PATIENT group is still required or empty. If there is a particular animal or group of animals as the subject of the testing the PATIENT group and the required PID segment must reflect that. In the case of actual animal testing it is much less likely than in the result message that the PATIENT group will be omitted in program testing because the laboratory needs this information for their accessioning information. Observations about the patient may become more variable in the order context. If specific findings about the patient are important for performance or interpretation of the requested testing, these findings should be communicated as OBX segments under the PID. The specific finding type should be coded as a LOINC code if at all possible; otherwise code system must be negotiated locally, in advance. For NAHLN or other program testing, specific finding codes will be provided.

## **SPECIMEN Group**

Things begin to get a little more interesting when we get to the specimen. Everything you know about the specimen still applies, except those details that pertain to condition and appropriateness when shipped. A little more information is included to cover situations like when a specimen is split up into multiple containers. The container type is indicated to help the lab properly identify the submitted sample(s). Codes are still needed for this field.

We will devote a section later to the confusing issue of orders for referral testing transmitted from one laboratory to another and the way the specimen aliquots get identified.

### ***OBSERVATION REQUEST Group***

Most of this group will look very familiar from the result message. The order changes a little, and there is a new group to deal with the priority of the request. The familiar ORC segment is still used to carry the Placer Group Number (referral number for program testing) and the date/time of the order. This segment is now required. So even if you don't have a referral number and you don't need to send the date/time of the order, entering device, etc. you must send an ORC with the order control code, "NW" for new orders or "SC" for changes. Changes are an advanced messaging topic that will be addressed in future versions of this guide.

The observation request segment (OBR) is much the same as in the result message, but only has the placer order number. Parent order is used for lab-to-lab referral testing and will be discussed in the section on that later in this guide. A key change from the original result message is the "Reason for Study" field. This replaces the observation on the visit in the result message that established the "reason for visit" for the entire message. Here we specify the reason for study at the request level. In normal usage—especially for NAHLN program testing—the reason will be the same for all orders in a message.

Another key change from the result message comes in the OBR.4 Order Service Identifier. This is now a true "order" code. LOINC is very good at naming observations. It is much less complete at the order level. There will inevitably be panels and other types of orders that cannot be LOINC coded, even by requesting new codes. In these cases, order code sets must be negotiated locally in advance.

### ***TIMING Group***

This one segment group is used to carry a single, very important, coded value. This is the TQ1.4 Order Priority code. The code system for this field may vary. For foreign animal disease testing it will be the priority codes from VS Guidance 12001.1 (formerly VS memo 580.4). For routine testing, it may be either a SNOMED code or a local code negotiated in advance. Program testing will specify the coding system in program guidance.

## **Keeping Similar NAHLN messaging Concepts Straight**

Reading the *NAHLN 3.0 HL7 Implementation Guide* can make you feel like you've had one too many pan-galactic gargle blasters. "What is the difference between 'Entered By' and 'Verified By' or between 'Role Instance ID' and 'Role Person ID'?" Some of these differences can seem almost completely arbitrary. Yet it is important for interoperability

that they be used consistently by all participating laboratories. Part of the confusion is just unfamiliarity with the HL7 standard which could, of course, be overcome by everyone taking years' of HL7 training. But a significant part of the ambiguity is inherent in the standard itself and the way it has evolved over the last thirty years, and then been adopted and interpreted for animal health laboratory use.

In *The Hitchhiker's Guide to the Galaxy*, the wealthier citizens of the universe became dissatisfied with having to take whatever happened to have evolved on the planets they inhabited. This frustration gave rise to a whole industry centered on the planet Magrathea. Magrathea became the center of the, "custom-made luxury planet building" industry. Because these planets were carefully designed from the core up, they had none of the irritants of the older planets, such as "the climate wasn't quite right in the latter part of the afternoon, or the day was half an hour too long, or the sea was exactly the wrong shade of pink."

Similarly, frustrations with the ambiguities, duplication, and conflicts that have crept into HL7 have given rise to an effort to design a new "planet" from the core out. In the near future HL7 version 3 or Fast Healthcare Information Resources (FHIR) will be complete and ready for use in systems such as the NAHLN.<sup>10</sup> The biggest change from version 2 to version 3 is the existence of a carefully designed data model underlying every element and attribute of any message. So, from the message and the standard it will be possible to precisely infer the meaning of any piece of information.

For purposes of the NAHLN messaging system, however, we are left in the imperfect world of HL7 version 2. But all is not lost. The underlying HL7 message structure does provide the vast majority of the standardization we need in order to stay compatible with other national and international systems such as the National Health Information Infrastructure (NHII) and Public Health Information Network (PHIN). If the specific parts of the structure relating only to animal health laboratories include some decisions that are somewhat arbitrary, that is fine as long as we are consistent in their usage. It would be nice if field names were sufficient, but in some cases ambiguity remains.

Many of the concepts contained in the NAHLN result message and related messages have field names that are very similar to other related concepts. For example, if you want to know when the blood was drawn for a test, you don't want to accidentally read the time the test was performed at the lab. The message structure provides many "slots" into which relevant times can be placed. The choices of exact usage have been made by those familiar with the HL7 standard in consultation with other experts at HL7 and with approval by the NAHLN IT Committee. It is hoped that other, related animal health information systems will adopt similar usage. This chapter addresses a few of the commonly confused concepts. It will help you find the right segments and fields to populate with specific information you need to send.

---

<sup>10</sup> Or so the story goes 23 years into the development of HL7 version 3. It turns out that designing a second system from the "core out" is harder than it sounds. HL7 version 3 never really took off for messaging. It remains at the heart of another type of interoperability standard called Clinical Document Architecture and now FHIR.

## Addresses

In *The Hitchhiker's Guide to the Galaxy*, the spaceship *Heart of Gold* is powered by an infinite improbability drive [see discussion under *Statuses*]. Because the ship travels at once through every point in the universe, one can use the *Heart of Gold* to travel anywhere instantly. The trick, of course, is to get it to stop at the right point. If you are looking for a known point such as the former location of the insignificant—and now vaporized—planet known as “Earth,” you can use a simple address such as, “Section ZZ<sub>9</sub> Plural Z Alpha.” If, on the other hand, you are looking for an unknown planet such as Magrathea, you must resort to more complex means of navigation such as improbability-maximizing algorithms.

Navigating our messages through the computer network cloud involves similar, but much less improbable issues. HL7 messages may travel across the building or across the world. Historically, across the building—or even across the room—was the normal use such as linking a laboratory instrument to the lab's LIMS. In those cases, the message must be addressed from one application to another. In the case of the NAHLN, most of our messages travel across the country. To do so, they must be addressed from one facility to another. In some cases both facility and application are needed to properly route the message from one place to another. Return addresses are critical for handling returned messages (acknowledgements), responses, etc. It may be helpful to think of the Message Segment Header (MSH) as an envelope used for sending the message. In the case of the NAHLN, this would be a regular postal mail envelope.

### From:

```
<MSH.4>
  <HD.1>{Your NPN [or PIN]}</HD.1>
  <HD.2>2.16.840.1.113883.3.5.6.1.4[.1]</HD.2>
  <HD.3>ISO</HD.3>
</MSH.4>
```

### Attn:

```
<MSH.3>
  <HD.1>VISUALAB</HD.1>
  <HD.2>2.16.840.1.113883.3.5.1.19.2</HD.2>
  <HD.3>ISO</HD.3>
</MSH.3>
```

### Contents:

```
<MSH.9>
  <MSG.1>OPU</MSG.1>
  <MSG.2>R25</MSG.2>
  <MSG.3>OPU_R25</MSG.3>
</MSH.9>
```

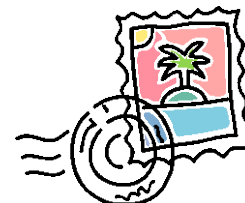
Optional: Only needed if more than one possible recipient at this address.

### Postmark:

```
<MSH.7>200609261100</MSH.7>
```

### USPS Tracking Number:

```
<MSH.10>9345AB</MSH.10>
```



### To:

```
<MSH.6>
  <HD.1>0034P2K</HD.1>
  <HD.2>2.16.840.1.113883.3.5.6.1.4</HD.2>
  <HD.3>ISO</HD.3>
</MSH.6>
```

### Attn:

```
<MSH.5>
  <HD.1>NAHLN</HD.1>
  <HD.2>2.16.840.1.113883.3.5.1.8</HD.2>
  <HD.3>ISO</HD.3>
</MSH.5>
```

Looking at this envelope you'll notice that the *From* and *To* addresses are for the facilities sending and receiving the message. Just as with normal mail, there may or may not need to be a block that directs the mail to a specific person—or application—within this organization. In NAHLN usage, the identifiers for sending and receiving facilities are ADT Non-producer Participant Numbers (NPN) or Premises Identifiers (PIN). The

OID identifies the ADT premises identification system that issued these NPNs. If PINs are used instead of NPNs, the last digit of the OID becomes a “1” instead of a “4”. The applications identified in the “attention” blocks are generated internally at each facility and defined by OIDs in each facility’s OID scheme. [See OIDs Vey!]

Other useful information carried along with the addresses includes the date and time sent, which is comparable to the postmark; and the message control ID, which is comparable to the tracking number for certified mail, UPS, FedEx, etc. Finally, unlike most junk mail, the message identifier tells the receiver what kind of contents to expect to find inside this envelope.

The more traditional use of HL7 within an institution is more like an inter-office distribution envelope. Some of the metaphors are not quite as accurate. Inter-office mail doesn't usually have a postmark or tracking number, for example.

### For Internal Company Use:

```
<MSH.4>
  <HD.1>{Your PIN}</HD.1>
  <HD.2>2.16.840.1.113883.3.5.6.1.1</HD.2>
  <HD.3>ISO</HD.3>
</MSH.4>
```

### From:

```
<MSH.3>
  <HD.1>VISUALAB</HD.1>
  <HD.2>2.16.840.1.113883.3.5.1.19.2</HD.2>
  <HD.3>ISO</HD.3>
</MSH.3>
```

### Contents:

```
<MSH.9>
  <MSG.1>OPU</MSG.1>
  <MSG.2>R25</MSG.2>
  <MSG.3>OPU_R25</MSG.3>
</MSH.9>
```

MSH.4 and MSH.6 Same  
value and both optional  
in this setting

### INTEROFFICE MAIL

### Sent:

```
<MSH.7>200609261100</MSH.7>
```

### Tracking Number:

```
<MSH.10>9345AB</MSH.10>
```

### To:

```
<MSH.5>
  <HD.1>Intelliview Data Warehouse</HD.1>
  <HD.2>2.16.840.1.113883.3.5.1.19.2</HD.2>
  <HD.3>ISO</HD.3>
</MSH.5>
```

The interesting thing to note here is that the important addresses are those of the sending and receiving applications. Because this message is internal to a single facility, the sending and receiving facility blocks are optional. It may be just as easy to program the interface engine to populate these on all messages, both internal and external as to leave them out. (This example uses a PIN in MSH.4 with the OID for PINs. This is just to illustrate that either can be used.)

## Times

Time is a critical factor in space travel whether the problem is calculating the best moment to slingshot off one planet on the way to another or trying to catch the millisecond during which Ford Prefect found a hole in the fabric of space-time through which to make his and Arthur's escape. Time is also an important variable in epidemiology and disease control. It is used in analysis and in laboratory work-flow. As

a result, the NAHLN messaging system deals with a large number of date/times throughout the message structure. This section will clarify the usage of dates and times throughout the results message.

Dates and times are all transmitted in the International Standards Organization (ISO)<sup>11</sup> 8601 “Basic”<sup>12</sup> format. The format follows the pattern “YYYYMMDDHHmmSS.ssss+/-ZZZZ” where Y is year, M month, D day, H hour in 24 hour clock, m minute, S second, s fraction of second and Z time zone as offset from UTC (Greenwich Mean Time). This is a very complicated way of saying that it can express anything from a year to a millisecond and give some idea of precision by leaving off any characters that aren't significant. A simple date might read “17760704” if I didn't know what time the Declaration of Independence was signed.

Specimen based testing is almost always a measure of some state in the patient or environment at the point in time that the specimen is collected. If I freeze a serum sample and run a test a year later, I don't know much about the animal's health at the time when the test is run, but I do know about its health when the sample was drawn. For this reason, the most physiologically significant date/time is when the sample was drawn—or in the case of necropsy samples, when the animal died. Because different specimens in an accession may have been collected at different times or even different days, this date/time is at the level of the specimen and can be found in the Specimen segment in SPM.17, Specimen Collection Date/Time. **If available, this is the most important date/time to send.**

## Catch Some Z's

The Z for time zone comes from the radio alphabet “Zulu” used to indicate zero offset from Greenwich Mean (Universal or Zulu) Time. If that isn't much help try this:

| Zone          | Standard | Daylight   |
|---------------|----------|------------|
| Eastern       | -0500    | -0400      |
| Central       | -0600    | -0500      |
| Mountain      | -0700    | -0600      |
| Pacific       | -0800    | -0700      |
| Alaska        | -0900    | -0800      |
| Hawaii        | -1000    | NA         |
| Arizona       | -0700    | NA (except |
| Navajo Nation | -0600)   |            |

Because we are interested in when something happened rather than what the local time was, we need a way to convert to some universal time. The time zone information lets us do that.

Times should be formatted as the local time at the sending facility along with its time zone offset.

This is most important for things that include minutes and seconds, of course, but is good practice for all date/time fields.

On Unix-based systems, you can get the value for ZZZZ or the full date string with the commands

```
$ date +%z
```

```
$ date +%Y%m%d%H%M%S%z
```

Unfortunately, the DOS date command lacks that functionality.

Most database systems will also have a system variable for time zone. For example: Oracle uses SESSIONTIMEZONE. SQL Server is more complicated but has both getDate() and getUTCDate() to figure out the time zone.

<sup>11</sup> Some say this isn't really an acronym. It is based on the Greek word isos meaning equal. But because is it always written in caps and can be coerced into the English words used here, this makes a good mnemonic.

<sup>12</sup> ISO 8601 also specifies a more commonly used “Extended Format” that includes punctuation to make it easier for human reading. The terser basic format was used, presumably, because of concern for minimizing message size in the early days of HL7.

Unfortunately, many specimens arrive without adequate information on collection time. But the laboratory always knows when it received the specimen. So, the message requires the Specimen Received Date/Time in SPM.18. For observations that do not involve specimen collection, the physiologically significant observation date/time is sent in OBX.14 but this is rare in NAHLN messaging. Systems *may* replicate the value from the specimen collection date/time here to simplify programming if desired.

Once the specimen is in the laboratory, a workflow begins. The details are, of course, specific to each test. But in any case, there comes a time when the actual analysis is performed. This is the “Date/Time of Analysis.” For tests that take time on the machine, this is the time the test was setup rather than the time it is finished, etc.<sup>13</sup> It is an attribute of the individual observation that is one of potentially many that make up the ordered test. This is recorded in the Observation segment in OBX.19.

Rarely, if ever, will results come off an instrument and immediately be transmitted to the NAHLN messaging system. Normally, there are additional validation and approval steps before the results can be considered final. The time that this status is assigned is the “Result Status Change Time.” It is generally assigned to an order as a whole and is therefore found in the Observation Request segment in OBR.22.

“Stuff happens” as the saying goes. Sometimes results that have been sent need to be corrected. When this happens we need a way to indicate that the results represent a change rather than a new observation, and to say when the event that triggered this change took place. This information goes in the Common Order segment in ORC.9, Date/Time of Transaction.<sup>14</sup>

### **OBR “Observation Request”**

The Observation Request segment OBR can be confusing. It serves two purposes, only one of which is reflected in its name.

First, it holds the information about the request or lab order, such as what battery of tests was requested, what types of specimens were sent, and so on. In this capacity, the OBR in the result message is essentially just a copy of the one that arrived in the order message.

But the OBR is also a general header for groups of observations that make up a single “test.” The most commonly used example is a complete blood count. A CBC is thought of as a test but is made up of several individual counts of different blood cell types. Each type is an observation. The OBR groups these into a CBC.

Another key date is the date of birth for the animal(s) tested. Often this is an estimate or a mid-range for a group. Until a better mechanism is developed for expressing a probability distribution (see HL7 version 3) this is expressed by limiting the characters included in the date/time field. A group of animals born in July might have the date of birth field listed as “200507.” Because no day is provided we can tell that this is accurate only to about the month. Date of birth goes in the Patient Identification segment in PID.7.

<sup>13</sup> This is the traditional usage according to Dr. Case. The important point is that it is the time of the actual observation of the result, NOT the time it is finally recorded, entered into a database, verified, etc.

<sup>14</sup> Correction messages are not currently fully supported by the NAHLN but are a potential future enhancement.



A final key date/time is actually the first one that appears in the message, but the last one generated. This is the date/time that the message was sent. It appears in the Message Header segment in MSH.7, Date/Time of Message. It is likely to be the only such field to include seconds. Time zone is more important here than in other date/time fields, so this field will usually appear something like “20051225093028-0500.”

### Statutes

The space vessel *Heart of Gold* in *The Hitchhiker's Guide to the Galaxy* was powered by an “Infinite Improbability Drive.” Without getting into the technical details, the system worked by passing through a series of states of finite improbability and keeping track of these states until they could be combined to form an infinite improbability. As anyone who has read any quantum physics knows, it is not impossible for a particle to instantly relocate anywhere in the universe, just infinitely improbable. By building up an infinite store of improbability, the *Heart of Gold* could travel instantly to any and all points in the universe. And it all depended on nothing more than keeping track of lots of statutes.

The NAHLN messaging system is not yet capable of tracking enough statutes to allow intergalactic travel, but it still has to keep up with a bunch of them. Each order has a status, the results as a group have a status, and each individual observation has a status. In implementing NAHLN messaging in a laboratory, we need to keep all these statutes in mind. The task gets somewhat easier because only a few of these are included in the actual messages in more than one status. Other statutes are important for designing the triggers that result in messages being constructed and sent over the system.

Orders are created by some entity that we call a “placer.” They may be sent as new orders, partial orders, cancelled orders, or a large number of other types. The NAHLN currently only deals with two. As of now, only new orders are sent in either NAHLN request or response messages. Cancellation and other unusual situations are not yet handled by the NAHLN messaging system.<sup>15</sup> From time to time, however, it will be necessary to send a correction to results already submitted. This type of message constitutes a replacement order. One of the main purposes for the Common Order segment (ORC) is to carry this status in its first field ORC.1 Order Control. The NAHLN uses “NW” for new orders and for results that are not expected to have order data on the server prior to results, and “SC” for results that are only submitted after corresponding order data—whether from an HL7 order message or other means—are required to be present on the server prior to submission of results. “RO” is used for replacement results. “SC” stands for status change, meaning results are now available. “RO” means we are requesting that the order be replaced. Replacement “orders” are currently used in the NAHLN only for correcting results or the information in PID, etc. If the order itself is incorrect, alternate communication is needed as in the case of cancellations. Eventually, replacement orders may become part of the order/order response messaging.

---

<sup>15</sup> Just as the electronic version of *The Hitchhiker's Guide to the Galaxy* was constantly being updated, so is this more earth-bound document. In the near future, the Implementation Guide for Order and Response messages will be published. As the order message usage evolves, you will find more information on the statutes involved in an upcoming release of the *Hitchhiker's Guide to NAHLN Messaging*.

Once the laboratory receives the request, things start to get interesting. Conceptually an order may be accepted, rejected, held, delayed, etc., but for our purposes the order is received with the specimen(s) or is held until the specimens arrive. Then the tests that make up the order are performed. For complex orders, different observations will be completed at different times. Each completed test results in a state transition. Most of these result in waiting for the rest of the tests in the order. Others result in immediate action. The NAHLN messaging system only handles partial and completed orders at present. The fact that the order is complete is carried in the Common Order segment in ORC.5 Order Status and is “CM” for complete, or “A”<sup>16</sup> for “some but not all results available.” In the future, cases of replacement results will be noted with “RP” for “Replacement.”

Individual observations pass through various states as well. Results from an instrument may be considered partial if only some of the data are available, or preliminary if they still require professional review before finalization. Note that the word “preliminary” means different things to different people. In the context of HL7 result status it means any result value that has not been completely finalized. In other contexts, it means accessions that have not been completed. These states do not result in NAHLN messages in the current system. Once the observation result is finalized it is ready to send via the NAHLN messaging system. This status is sent in OBX.11 Observation Result Status which is virtually always “F” for final. The exception is when changes are required. In this case, “C” means that the value is a change. In a change message ORC.1 is “SC” (status change), and the OBX's are either “F” for those values that have not changed, or “C” for those that have. If a test cannot be performed, such as when a bad specimen has been submitted, the results are “cancelled.” This is sent as Observation Result Status “X.”

### **Identifiers**

The Sirius Cybernetics Corporation developed—and presumably patented—a concept of robotics known as GPP or “Genuine People Personalities.” The unfortunate result was Marvin the Manic-Depressive Robot. The NAHLN, wishing to avoid similar difficulties, chose to adopt cold and impersonal identifiers instead of friendly names for the various entities that make up a laboratory case.

Two clues combine to make each identity. The first is the position in the message. Each identifier field or component has a meaning defined by HL7 and further refined by the NAHLN. For example, if I look in SPM.2.1 I know what I'm seeing is the identification for the specimen as designated by the submitter (placer.) The second clue is who assigned the identifier in the first place.

### **How are they assigned?**

To be useful in a national database, these identifiers must in most cases be globally unique. But how can I be sure that the values I assign will not duplicate ones you assign? There are two ways this can be done. The first is what was adopted by the Internet for

---

<sup>16</sup> Don't ask me where they came up with some of these official HL7 abbreviations.

## The Hitchhiker's Guide to NAHLN Messaging

domain names or by the Animal Disease Traceability system for premises and animal identifiers. This solution requires everyone to go to a central authority to obtain a set of unique identifiers for its use. This works for relatively static systems like Premises ID's. But for things like specimen identifiers, we don't want to have to go back to some central place for every new number. (Some surveillance programs *do* assign specimen identifiers nationally at the program level, usually in the form of barcodes.)

The second approach uses a hierarchy. Every identifier is assigned by someone or some system. Your first name was given to you by your parents and recorded in a vital records system somewhere. If your first name is "John" that helps identify you. But there may be many "Johns" in even a small town. In some societies this is solved by adding the name of your father (who gave you the name) so that you might become "John Jackson" or "Ivan Mikhailovic." If there were only one "Jack" or "Mikhail" in the world, you would then have a universally unique name. For that to work, your father's name had to be unique to begin with. And, of course, this will only be true as long as your father didn't name two children with the same name, but we leave that responsibility to him.

Each entity can assign numbers on its own, ensuring that it doesn't duplicate any of its own identifiers, but not worrying about other entities. Such identifiers are said to be "locally unique." So, the California lab may assign a case number of "200512001A" and Colorado may also assign the number "200512001A." Each is a perfectly good case number within their labs. When they go to the NAHLN however, we get a collision. We can only tell them apart if we know *whose* case number "200512001A" we are looking at. To make a locally unique identifier into a globally unique identifier, we need a globally unique identifier for the entity that assigned the locally unique identifier. This globally unique identifier is assigned by some central authority as in the previous paragraph. The NAHLN uses Object Identifiers (OID's) as this globally unique identifier. The OID's are, in turn, issued by the American Association of Veterinary Laboratory Diagnosticians (AAVLD).

So, these two case numbers are easy to tell apart (if you look closely):

```
<PV1.19>
  <CX.1>200512001A</CX.1><!--Accession number-->
  <CX.4>
    <HD.2>2.16.840.1.113883.3.5.1.2.1</HD.2> <!-- OID representing a CAHFS accession number-->
    <HD.3>ISO</HD.3>
  </CX.4>
</PV1.19>

<PV1.19>
  <CX.1>200512001A</CX.1><!--Accession number-->
  <CX.4>
    <HD.2>2.16.840.1.113883.3.5.1.5.1</HD.2> <!-- OID representing a Colorado State accession number-->
    <HD.3>ISO</HD.3>
  </CX.4>
</PV1.19>
```

Identifiers in the NAHLN almost always have an explicitly identified “assigning authority” or standard coding system. The set of identifiers assigned by any one assigning authority forms a set of unique values known as a “namespace.” The list of identifiers for assigning authorities must itself constitute a namespace. The NAHLN uses two types of globally unique identifiers for assigning authorities or code systems. The International Organization for Standards (ISO) has a system of Object Identifiers (OID's) that are always unique. The NAHLN along with AAVLD manages a range of these identifiers for use by member labs and related identifier systems. For widely recognized coding systems such as LOINC and SNOMED, the HL7 standard provides abbreviations to identify the set of codes being used. See NAHLN Standard Terminology below for more details on using these predefined code sets.

### OID's

ISO Object Identifiers follow a pattern of numbers separated by dots. Each group subdivides the sequence before it (known as an “arc.”) By assigning arcs to be managed by various groups around the world, ISO is able to keep this system globally unique. The NAHLN arc is managed by AAVLD and is off of an arc managed by HL7.

2.16.840.1.113883.3.5 is the OID for AAVLD.

Each laboratory can obtain its own OID arc from AAVLD. It is then the lab's prerogative to assign arcs below it as it sees fit. Carefully assigning arcs can help divide up the job of creating and maintaining identifier systems for things like accession and sample identifiers.

Users never see OID's. But they are essential inside systems to keep all the identifier types straight.

The identifiers themselves may be a simple sequence of numbers or may have logic included in the code. They may or may not include checksums or other features to improve data integrity. The only real requirement is that each identifier be unique within its namespace. We commonly use the words “identifier” and “number” interchangeably in this context. Identifiers may be made up of digits, but we seldom use them as actual numbers so this is a figure of speech. For some systems such as most databases, an easy way to generate unique identifiers is to have the system assign sequential numbers. Once assigned, all we care about is the uniqueness. The numerical value is irrelevant.

### Who is responsible to assign them?

Some identifiers used within the NAHLN are assigned on a national level. The Animal Disease Traceability system (ADT) Animal Identification Numbers (AIN's) and Premises Identification Numbers (PIN's) are examples of nationally assigned identifiers.

Other identifiers are assigned on a program level. The USDA's Emergency Management and Response System (EMRS) assigns a number of identifiers for various pieces of information including referral numbers for investigations.

Many identifiers are assigned at the laboratory level. These include accession numbers, sample identifiers, order identifiers and result identifiers. In many cases, these identifiers may not be shown to the users of software such as the LIMS, but usually exist internally. The lab's OID for this type of identifier is used as the assigning authority to make clear that the lab assigned the identifier and to make sure it can be differentiated from the same identifier assigned by another lab. (Such duplication is called an identifier "collision.")

This basic structure is repeated throughout the NAHLN messaging system wherever globally unique identifiers are required.

There are only a couple of cases when identifiers may not be unique at all. Animal identifiers assigned by owners or other outside agents may be simple strings like "Y\_123." That is a very different situation from official animal identifiers such as the National Uniform Eartag System (NUES) tag numbers. Specimen identifiers coming in from non-program sources also probably have very simple specimen labels such as "1, 2, 3" rather than unique IDs. We note that these are not necessarily unique by omitting the assigning authority completely.

## Hierarchic Designators (HD)

For the most part this guide tries to stay away from the fine-grained details of HL7 data types and other message structures. The Implementation Guide is the place to go for that. But one data type in particular seems to cause confusion. The Hierarchic Designator (HD) is used to provide a universally unique identifier either on its own or when used as an "Assigning Authority." It is the "or" in that last sentence that causes the problem. Let's look at the various usages.

The HD datatype consists of three components: a namespace identifier, a universal identifier, and a universal identifier type. In the NAHLN, the last two are fairly straightforward. The universal identifier is always an OID [see above] and thus, the universal identifier type is always "ISO". The confusion comes about in what the universal identifier identifies. The HL7 standard provides for several different usages of HD that

### OIDs Vey!

#### Optional Simplified OID Scheme

When OIDs are used to identify the assigning authority of identifiers the intent is that the identifier plus OID combination will be globally unique. Thus, the ideal lab OID setup is to have one OID for each system of identifiers. This can result in a large number of OIDs for accession numbers, order numbers, sample numbers, client numbers, etc.

Because the NAHLN keeps all identifiers in context there is no risk of confusing sample numbers with order numbers, etc. So, an optional way of assigning OIDs is to use a single OID—possibly even the lab's root OID—as the assigning authority for *all* identifiers assigned by the lab. Additional OIDs would only be required if a system-change causes numbering to start over or if a lab uses more than one, non-mutually-unique system for the same type of IDs (two LIMS systems, etc.).

This simplified OID scheme that started out as an optional alternative has become the normal method for all but the most complex laboratories. Only labs that may have multiple systems assigning multiple identifier types have tended to use the more complete scheme.

(Unique OIDs are still needed for lab specific method codes, etc.)

have evolved over the years starting with HD's origin as a simple application identifier. This historical baggage is part of the confusion.

In the NAHLN messaging system, the universal identifier is always an assigning authority of another identifier. It may be the assigning authority of the first component of the HD. This usage is seen in the Message Header in MSH.3-6. In all other cases the universal identifier is the assigning authority of some other identifier where the HD itself fills an "assigning authority" field in a segment. This is also the usage when HD appears disguised as the last three components of an Entity Identifier (EI) data type. In each of these cases the namespace identifier is left empty.

These two, slightly different, usages are dictated by the NAHLN Message design and are always the same in a given location in the message. A sending application must use the HD in the manner prescribed for a given field, and the receiving application does not need to attempt to differentiate between usages. Except...

In rare cases, an identifier may have been issued by a known entity that has no universal identifier (OID). This may be the case with farm tags, veterinarian submitted specimen identifiers when the specimens were not collected as part of a disease control program and so on. In these cases only the namespace identifier may be populated in an assigning authority field. In these unusual cases, the receiving application cannot assume universal uniqueness of the identifier/assigning authority pair. The information may, however, be usefully used by human beings attempting to trace back the identifier, so they should be retained. As noted above, the assigning authority is omitted when the identifier is not known to be unique in *any* namespace.

So, what are some of these things that need to be globally identified and how are these represented in the message?

### Key Identifiers and Why We Need Them

The chief scribe of this guide has a very common name and, it seems, a name shared with at least one nefarious bad-guy. He used to always end up on the no-fly list. From time to time he would be taken into custody by Immigration when trying to return home from overseas. Each of these incidents was eventually resolved through a time-consuming process involving some human officer conducting an investigation/interrogation to differentiate this individual from the one they were really after. What was the answer? As you might guess from the title of this chapter, it was identifiers. Homeland security can issue an identifier they call a "redress number." Customs and Border Protection has a program called "Global Entry" that assigns an identifier specifically to ease entry to the country. Each of these is linked to passport numbers, driver's license numbers, airline frequent flier numbers, and on and on. The point being that with the right identifiers, even the Department of Homeland Security's computers can *usually* tell this human from the human they are looking for. Similarly, with the right identifiers in messages, the NAHLN's computers can correctly deal with more situations without having to use time-consuming human processing. Let's look at a few of the key identifiers and why they

may be important in processing NAHLN messages even though humans would seldom look at them.

The Message Control Identifier (MSH.10) is the main identifier that allows reference to a specific message. The number must be unique for each lab. You would use the same MSH.10 value only if you needed to resend the exact same message such as when a network failure interrupts the actual transmission process. A human might simply look up all the messages for a given accession. The computer needs something more specific. The Message Control Identifier is returned to the lab in the acknowledgement message so the lab can find the specific message referenced.

Premises Identifiers including PINs and LIDs--but *not* state location identifiers--sent in the PREM Role segment (ROL.13/PL.10/EI.1) are essential for assigning laboratory results to the appropriate location. This may mean giving the correct farm credit for herd certification testing, assigning results to the correct premises in an outbreak investigation, etc. A human could probably work from premises name and address, etc. but for reliable computer processing we need a true identifier.

Patient Identifier(s) are pretty obvious but correct representation is critical. These are used for any number of things that we really want the computer to do for us. To find prior testing on an animal from a tag number read years later we need to be able to search on official animal identifier. Barn names, farm tags, etc. may be helpful in some cases once a human is back in the loop, but a correctly formatted animal identifier is essential for finding the record in the first place. Any number of identifiers for the same animal can go in PID.3/CX.1. If the patient is a flock the identifiers are for the group, *not* a list of individual animal IDs (see Patients, Samples, and Specimens).

In nearly every case, we need to know which specimen has been tested. Specimen Identifiers in (SPM.2) come in two flavors. One type of ID is assigned by the submitter. This may be a true unique ID such as a barcode or it may be a simple sequence (1, 2, 3, etc.) This is the most important identifier for humans. But it is often not unique enough for the computer to know how to tell one specimen from all others, count distinct specimens tested, etc. For this global use, we need a true unique identifier assigned by the laboratory. The submitter (placer) identifier goes in SPM.2/EIP.1. The lab's (filler) identifier goes in SPM.2/EIP.2. Each is essential for its own purposes.

Is this the same test or a new one? This seems like a simple question but without an identifier for the test, how can the computer answer it? What most of us mean when we say, "test" is an order along with one or more results. So to the computer a "test" identifier is actually an Order Identifier. As with specimens, the submitter *may* provide a unique order number from their system (OBR.2) but in current NAHLN usage with no electronic order there is usually no placer order number. But the lab always has a unique identifier for the order although this may never appear to users. This unique identifier goes in the placer order number (OBR.3) (In other HL7 messages this may appear in ORC.3 with the same semantics.) This identifier becomes absolutely essential if a lab ever needs to send an update, correction, or completion of a "test". An example is if the

demographics are incomplete even as the first results become available. The lab may need to be able to tell the NAHLN to update the “order” with the complete demographics--such as the PIN required for payment! The only way the lab can tell NAHLN what order to update is with the filler order number.

While the Laboratory Messaging System used by the NAHLN looks, to the labs, like a “black box”, it does a lot of specific routing of results to program-specific databases. To properly assign results to programs the computer needs a Program Identifier. The NAHLN uses OIDs for programs. These can be found in the AAVLD OID registry under “AHERMS Programs.” The Program Identifier OID goes in ORC.4/EI.3. A slightly unusual feature is that in cases where the program issues a Referral Number, the Program Identifier OID doubles as the assigner of the Referral Number. An even more far-flung use of the program OID is when the program issues barcode specimen IDs. In that case the program OID is also the assigner of the placer specimen identifier. But the semantics of this use are completely different from those of ORC.4. It is risky for the computer to “guess” that if a barcode was issued, it must have been issued by the testing program. The Program Identifier OID must always be provided in ORC.4/EI.3.

Finally, even individual results need unique identifiers. Think about three different situations. A lab sends a result to NAHLN. Then one of three things may happen. For convenience the same result may be resent along with other information, additional results, etc. The NAHLN needs a way to know for sure that this is *not* a new result. Or, perhaps the lab discovers that they entered an incorrect value and needs to correct the original result. Finally, imagine that the lab has reason to doubt the result and retests the same sample as part of the same order. This is a *new* result. In the first two cases, the lab needs to send *the same* result identifier. In the last case they need to send a *new* result identifier. LIMS systems essentially never show users a unique identifier for each result even though their database functions require one internally. For years we tried to find ways to infer the difference between these three cases without a unique result identifier. The addition of a unique result identifier in OBX.21 solves this problem once and for all.

### **Actors and Roles**

Keeping straight who's who in an HL7 message can be confusing. And then, when you think you have it figured out, it turns out that the lab mice were actually in charge the whole time. To make matters even more confusing one person may have multiple identifiers. In *The Hitchhiker's Guide to the Galaxy*, Ford Prefect's original name turns out to be unpronounceable except in an obscure Betelgeusian dialect. To allow people to say his name, he adopts the name of a model of car. So, in parts of Betelgeuse he is known by one name, and everywhere else by another. In this section we look at some of the many segments and fields that deal with people and the roles they play in ordering, submitting, collecting, running, and reporting tests.



## Responsible Diagnostician

At the highest level a laboratory submission can be seen as an interaction between a responsible party in the field who submits a request for testing to the laboratory, and a laboratory diagnostician who supervises the conduct and interpretation of one or more tests on one or more samples. All this takes place in the context of a laboratory accession. Because HL7 was originally concerned with hospital information systems, the central event context is called a “patient visit.” But for the NAHLN, the accession is the *visit*.

With that usage in mind, it makes sense that the “Attending Doctor” is the Laboratory Diagnostician responsible for the accession. This role is often known as the “Case Coordinator.” Information about this person is found in the field PV1.7. The case coordinator is identified both by name (in PV1.7.2-5) and by a coded identifier in PV1.7.1. As with all NAHLN identifiers, we need to know who assigned the ID. This information is found in PV1.7.9.

```
<PV1.7>
  <!--Attending doctor.
    For the NAHLN it represents the responsible diagnostician for the submission. AKA Case coordinator.-->
  <XCN.1>
    <!--Case coordinator ID-->
    VET001
  </XCN.1>
  <XCN.2>
    <!--Case coordinator name-->
    <FN.1>Vet</FN.1>
  </XCN.2>
  <XCN.3>Joe</XCN.3>
  <XCN.4>B.</XCN.4>
  <XCN.9>
    <!--OID for Lab, representing the assigning authority for the ID in XCN.1-->
    <HD.2>2.16.840.1.113883.3.5.1.2</HD.2>
    <HD.3>ISO</HD.3>
  </XCN.9>
</PV1.7>
```

## Submitter

That leaves the submitter. In foreign animal disease control programs the submitter is a specific role assigned by USDA. This information goes into a Role (ROL) segment that follows the patient visit information. We use two distinct sets of identifiers within the Role segment. The Role Instance ID (ROL.1) may be an identifier assigned by USDA for veterinarians submitting samples for FAD testing. The same person is normally known to the laboratory by name and with the laboratory's own client identifier. This laboratory assigned identifier is assigned to the Role Person (ROL.4) field. Because the Role segment can be used for different things, we indicate in the Role code (ROL.3) that this is the *submitter* role “SUB.”

## The Hitchhiker's Guide to NAHLN Messaging

```
<ROL>
<!--The first ROL segment must define the submitter of the samples-->
<ROL.1>
  <!--Role Instance ID-->
  <EI.1>
    232345<!--The ID of the submitter as assigned by the authority listed in EI.3-->
  </EI.1>
  <EI.3>
    2.16.840.1.113883.3.5.6.2 <!--OID for USDA PersonID-->
  </EI.3>
  <EI.4>ISO</EI.4>
</ROL.1>
<ROL.2>UC</ROL.2><!--Code for unchanged. No changes are requested for these ROL data-->
<ROL.3>
  <!--The Role being played by the person or organization represented by this segment-->
  <CWE.1>SUB</CWE.1>
  <CWE.2>Submitter</CWE.2>
  <CWE.3>HL70443</CWE.3>
</ROL.3>
<ROL.4>
  <!--The person or organization in this role.-->
  <XCN.1>
    <!--The ID of the person or organization as assigned by the authority listed in XCN.9.
    This would usually be supplied by the filler (the receiving laboratory)-->
    GIB001
  </XCN.1>
  <XCN.2>
    <FN.1>Gibson</FN.1>
  </XCN.2>
  <XCN.3>Henry</XCN.3>
  <XCN.4>T. </XCN.4>
  <XCN.5>Jr</XCN.5>
  <XCN.9>
    <!--This is the OID representing the assigning authority for the ID in ROL.4; a Lab in this case.
    This will not be sent if the ROL.4.1 (ID) is not provided -->
    <HD.2>2.16.840.1.113883.3.5.1.2</HD.2>
    <HD.3>ISO</HD.3>
  </XCN.9>
</ROL.4>
...
</ROL>
```

The role segment continues with additional information such as address and phone number of the submitter.

### Collector

The specimen collector is handled in exactly the same way as the Submitter except for the role code that for the collector is “COL.”

```
<ROL.3>
  <!--The Role being played by the person or organization represented by this segment-->
  <CWE.1>COL</CWE.1>
  <CWE.2>Collector</CWE.2>
  <CWE.3>HL70443</CWE.3>
</ROL.3>
```

### Attending Veterinarian

And the attending veterinarian, if appropriate, is handled in the same way again with a role code of “AT.”

```
<ROL.3>
<!--The Role being played by the person or organization represented by this segment-->
<CWE.1>AT</CWE.1>
<CWE.2>Attending</CWE.2>
<CWE.3>HL70443</CWE.3>
</ROL.3>
```

## Payor

More important in the Order message, the Payor role—code PAY—is needed to carry the entity paying for testing in this accession. ROL.1 is used to carry the billing account identifier.

## Premises Role (An Odd Variation)

The normal use of the ROL (role) segment is to express the fact that some person played a role participating in the event being reported—lab results in this case. Part of defining this *participation* is information about where the participation took place. Once in a while, we need to say that the participation took place, but not precisely by whom. This is the HL7 equivalent of a passive voice sentence. “The specimen was collected on Joe’s Farm.” The actual subject noun is missing from this type of sentence. The interesting thing is that the place the specimen was collected rather than the person doing the collecting. (In our message, that is sent elsewhere.) The same thing happens when using a ROL segment to send place information. The actor part is left out or left essentially blank.

The role code used for recording premises is PREM. The actual premises information from the Animal Disease Traceability system (ADT) goes in the ROL.13 (Person’s Location) field. The location type code is the ADT code for operation type. The location description is generally the “Entity Name” field from the ADT premises data. The actual premises identifier goes in the Comprehensive Location Identifier component.

```
<ROL.13>
<PL.6>P</PL.6>
<PL.9>Joe's Farm</PL.9>
<PL.10>
  <EI.1>000UDC0</EI.1>
  <EI.3>2.16.840.1.113883.3.5.6.1.1</EI.3>
  <EI.4>ISO</EI.4>
</PL.10>
</ROL.13>
```

Note that the “Other Geographic Designation” field in the address (XAD.8) has changed usage since the early prototypes of the NAHLN message. This was originally planned to be the place to store the ADT premises ID. After further study showed that this really belonged in the Comprehensive Location Identifier, the Other Geographic Designation field reverted to its more normal HL7 usage for things like latitude and longitude.

In version 2, HL7 does not have a structured way of sending geographic coordinates. These are sent as one string with parsing left up to the implementation. Because no current NAHLN usage relies upon this, and most LIMSs don’t even maintain it, this is

## The Hitchhiker's Guide to NAHLN Messaging

left as a place to send a human-readable set of geo-coordinates if an implementation needs to. It is an RE field with the emphasis on the “or Empty” part. It will rarely be sent.

So, a complete premises ROL segment might look like this.

```
<ROL>
  <ROL.2>UC</ROL.2>
  <ROL.3>
    <CNE.1>PREM</CNE.1>
    <CNE.2>Animal Origin Premises</CNE.2>
    <CNE.3>L</CNE.3>
  </ROL.3>
  <ROL.4>
    <XCN.2>
      <FN.1>Not Applicable</FN.1>
    </XCN.2>
  </ROL.4>
  <ROL.11>
    <XAD.1>
      <SAD.1>123 Pasture Lane</SAD.1>
    </XAD.1>
    <XAD.2>Suite 10</XAD.2>
    <XAD.3>Farmtown</XAD.3>
    <XAD.4>SC</XAD.4>
    <XAD.5>28888</XAD.5>
    <XAD.8>34.1234N, 81.1234W</XAD.8>
  </ROL.11>
  <ROL.13>
    <PL.6>P</PL.6>
    <PL.9>Second to the last farm on the road</PL.9>
    <PL.10>
      <EI.1>000UDC0</EI.1>
      <EI.3>2.16.840.1.113883.3.5.6.1.1</EI.3>
      <EI.4>ISO</EI.4>
    </PL.10>
  </ROL.13>
</ROL>
```

OK, but what if all you know is the state the specimen came from? This may apply in the case of some anonymous testing, etc. Then you send a very abbreviated version of the same thing. The NAHLN message tries to have a place for as much as you know but to accept as a minimum what is absolutely required.

```
<ROL>
  <ROL.2>UC</ROL.2>
  <ROL.3>
    <CNE.1>PREM</CNE.1>
    <CNE.2>Animal Origin Premises</CNE.2>
    <CNE.3>L</CNE.3>
  </ROL.3>
  <ROL.4>
    <XCN.2>
      <FN.1>Not Applicable</FN.1>
    </XCN.2>
  </ROL.4>
  <ROL.11>
    <XAD.4>SC</XAD.4>
  </ROL.11>
</ROL>
```

Finally, what if you don't even know what state the specimen came from? Sorry, you need to get over it and start collecting this information.<sup>17</sup> The NAHLN/LMS system can't hope to do its job as an information router and analysis source without at least this minimal geographic source information. This may require business process changes for some labs.

### Data Enterer and Verifier

Two more people within the lab need to be identified in the NAHLN message. The person who enters the results into the computer system is the “Data Enterer,” and the person who provides the final validation of the results is the “Data Verifier.” They are identified by laboratory assigned identifiers and listed in the Common Order segment (ORC) in ORC.10 and ORC.11 respectively.

### Owner

The owner of the location or animals being tested is also important in many cases. Because human hospital patients obviously don't have owners, you will not find a reference to “owner” in the HL7 standards. That is just as well. We really don't want to get into the whole fuss over whether owners are “owners” or “guardians” or whatever. The reason we need to know the owner is the same as the reason hospitals need to know the patient's next of kin. And it turns out that the Next of Kin (NK1) segment is the perfect place to transmit owner or other responsible party information.

Note: When sending this required NK1 segment when owner information is not allowed to be sent (as in some state surveillance programs), use the following structure:

```
<NK1>
  <NK1.1>1</NK1.1>
  <NK1.3>
    <CNE.1>OWN</CNE.1>
    <CNE.2>Owner</CNE.2>
    <CNE.3>HL70063</CNE.3>
  </NK1.3>
</NK1>
```

Or even:

```
<NK1>
  <NK1.1>1</NK1.1>
</NK1>
```

When more complete owner information is available, the segment is more completely populated as in the following example:

---

<sup>17</sup> A very rare exception was made for an anonymous swine influenza data collection project.

## The Hitchhiker's Guide to NAHLN Messaging

```
<NK1><!--Next-of kin segment. Used to represent the animal owner or responsible party-->
<NK1.1>1</NK1.1>
<NK1.2>
  <!--Name of owner-->
  <XPN.1>
    <FN.1>Smith</FN.1><!--Last name. -->
  </XPN.1>
</NK1.2>
<NK1.3><!-- The relationship of this NK1 to the animal or specimens. In this profile it will always be OWN-->
  <CWE.1>OWN</CWE.1>
  <CWE.2>Owner</CWE.2>
  <CWE.3>HL70063</CWE.3>
</NK1.3>
<NK1.4><!--The address of the owner.-->
  <XAD.1>
    <SAD.1>348735 Laguna Pl. </SAD.1>
  </XAD.1>
  <XAD.3>Agtown</XAD.3>
  <XAD.4>CA</XAD.4>
  <XAD.5>95999</XAD.5>
</NK1.4>
<NK1.5><!--Home phone number of the owner. This section may repeat.-->
  <XTN.2>PRN</XTN.2>
  <XTN.3>PH</XTN.3>
  <XTN.12>555-444-4444</XTN.12>
</NK1.5>
<NK1.6><!--Business phone number of the owner. This field may repeat.-->
  <XTN.2>WPN</XTN.2>
  <XTN.3>PH</XTN.3>
  <XTN.12>555-333-3333</XTN.12>
</NK1.6>
<NK1.20>
  <CWE.1>eng</CWE.1>
  <CWE.2>English</CWE.2>
  <CWE.3>ISO639</CWE.3>
</NK1.20>
</NK1>
```

### ***Patients, Samples, Specimens***

In the human medical world, the word “patient” has a simple meaning. It is the individual human being treated or receiving preventive care. Veterinary medicine is not so straight forward. Our patients may be any species instead of just one. The object of our care is often not even an individual animal. In poultry, swine, and aquaculture medicine, the “patient” is the entire group. If such groups stay together throughout the production system, the ADT allows them to be identified by Group Identification Numbers (GIN) rather than individual animal identifiers. Other groups may consist of individually identifiable animals, but nevertheless be diagnosed, and treated as a group. The NAHLN messaging system is designed to capture the appropriate level of identification for animals, groups, and the samples that are drawn from them. A few basic guidelines apply.

First, if any trace-back, epidemiologic analysis, etc., needs to be done on an individual animal basis then each animal needs to have its individual identity tracked through the system. Each needs its own Accession Detail with its own PID segment and animal identifier(s). Ideally this identifier list includes an ADT AIN, but in any case includes the best individual ID available.

For cases where the group is to be treated as a single unit for all epidemiology, a single PID segment represents the entire group. The identifier list should include the ADT GIN if available or other group ID. In this case each specimen is considered to be a sample from the group. In these cases, the group is entirely analogous to a single patient. In other cases the group is conceptually made up of distinct but anonymous individuals. For example, we might not care *which* animals are affected but do care how many different animals test positive. Care is needed to be sure we can track these anonymous animals' specimens and results.

If two or more specimens need to be identified as being from the same anonymous animal, they may receive the same Placer Assigned Specimen Identifier—and the same bar code. The individual specimens are then differentiated by their specimen type and the locations from which they were taken. This is a variation from the normal usage in that the specimen ID is no longer a valid primary key onto the list of specimens. Instead, a compound key made up of the specimen identifier combined with the specimen type and source site is needed to uniquely identify each sample record. Some LIMS handle specimen identification the same way. For LIMS that assign a unique identifier for each individual specimen, this is not a problem, but if multiple specimens are given the same identifier, it is crucial to keep the associated data (i.e. specimen type and source site) in order to assign the correct result to the correct sample in the message.<sup>18</sup>

### **Unique Specimen Identifiers**

The most fundamental function of a laboratory is to receive specimens, perform testing on them, and return the results to the requester. Identity of the specimen is, therefore, a key element in any message. This identifier may take any of several forms.

For program testing, the specimens are sometimes uniquely identified at the program level using barcodes. The value coded in each barcode is unique within the issuing program. Normally, pairs or more of the same barcode are printed so that one can be placed on the specimen and another on the request form. This is the prototype NAHLN/LMS form. The barcode value goes in the Placer Specimen ID Identifier field and the program's OID in the Assigning Authority field.

```
<SPM.2>
  <EIP.1>
    <EI.1>A1004324</EI.1>
    <EI.3>2.16.840.1.113883.3.5.8.4.1</EI.3>
    <EI.4>ISO</EI.3>
  </EIP.1>
</SPM.2>
```

In a perfect world everyone—and everything—has a name a name. And unlike poor Ford Prefect, the name is pronounceable in languages other than an obscure Betelgeusian dialect. But some of our specimens are not so lucky.

---

<sup>18</sup> This paragraph is ugly. We know. But it was necessary to deal with the way the Scrapie program does business in the "real-world." It does work as long as all the pieces are performed correctly. It creates problems if, for example, a submitter misidentifies a specimen type.



A "Ford Prefect" (WikiMedia)

This *Ford Prefect* has a perfectly good Specimen ID, "741 DJH". But some of the specimens that come into our laboratories arrive with absolutely no identifying label. To be sure we can tell them from an automobile marketed in the United Kingdom, we need to make sure they get some kind of unique identifier. In other cases, the label on the specimen may be in no way globally unique. It may simply be a serial number, "1, 2, 3, etc."

The NAHLN message has a place for the laboratory's internal identifier for the specimen. Virtually every LIMS has such an identifier, even if they don't show it to the user. In some it is a concatenation of the accession number with a sequence or the above serial number, in others it is an integer key or even a GUID. This identifier, combined with the lab's OID, is globally unique.

When submissions come directly from industry veterinarians rather than program specimen collection activities they will not come in with globally unique specimen IDs. Instead labs will need to report whatever ID did come on the submission along with a unique Filler Specimen ID assigned by the lab. This unique ID may have to be constructed from the accession number plus a specimen sequence, etc., if the LIMS does not make a system-unique ID available. The important unique ID is then the Filler Specimen ID (EIP.2) and it needs the lab's OID (or Lab's LIMS OID or Lab's specimen ID OID as appropriate) in EI.3 and "ISO" in EI.4.



```
<SPM.2>
  <EIP.1>
    <EI.1>123</EI.1>
  </EIP.1>
  <EIP.2>
    <EI.1>C14001234-1</EI.1>
    <EI.3>2.16.840.1.113883.3.5.1.9</EI.3>
    <EI.4>ISO</EI.4>
  </EIP.2>
</SPM.2>
```

If no identifier exists on the specimen as it arrives, SPM.2/EIP.1 Placer Specimen ID may be omitted entirely, but because the placer specimen ID is used in human-readable presentations of the messages, it is sometimes important that this field include *something*. In those cases, the sending system may repeat the Filler ID as if it was the placer ID. While this sounds wrong, the lab's OID in EI.3 will make clear that the “Placer ID” was actually provided by the laboratory as the specimen was received.

```
<SPM.2>
  <EIP.1>
    <EI.1>C14001234-1</EI.1>
    <EI.3>2.16.840.1.113883.3.5.1.9</EI.3>
    <EI.4>ISO</EI.3>
  </EIP.1>
  <EIP.2>
    <EI.1>C14001234-1</EI.1>
    <EI.3>2.16.840.1.113883.3.5.1.9</EI.3>
    <EI.4>ISO</EI.4>
  </EIP.2>
</SPM.2>
```

Some specimens will have unique identifiers in both Placer Specimen ID and Filler Specimen ID.

```
<SPM.2>
  <EIP.1>
    <EI.1>A1004324</EI.1>
    <EI.3>2.16.840.1.113883.3.5.8.4.1</EI.3>
    <EI.4>ISO</EI.3>
  </EIP.1>
  <EIP.2>
    <EI.1>C14001234-1</EI.1>
    <EI.3>2.16.840.1.113883.3.5.1.9</EI.3>
    <EI.4>ISO</EI.4>
  </EIP.2>
</SPM.2>
```

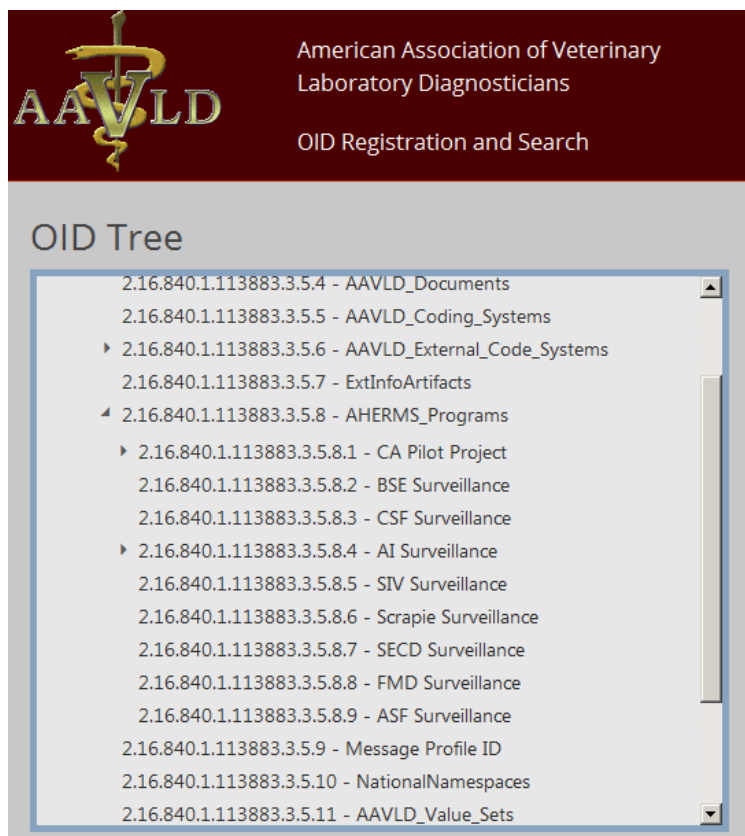
The laboratory's unique specimen identifier is *always* sent in EIP.2 and then send the best available placer identifier in EIP.1 as described above.

## Program Identifiers

A common question besides “why was this test run” is “what surveillance or response program is this test for?” USDA and states have a large number of active and passive surveillance programs as well as systems to manage emergency events. For purposes of

## The Hitchhiker's Guide to NAHLN Messaging

the NAHLN each of the federal programs is issued an OID under the general category "AHERMS Programs."



These OIDs are used to identify the program as the issuer of program specific barcodes, referral numbers, etc. So, in a typical message, you might find one or more AHERMS Program OIDs in several places. Each has its own meaning. In the SPECIMEN group, as an assigning authority of one of the specimen identifiers in SPM.2 or SPM.3, it means the identifier namespace is managed by the program. The program might tell collectors to use a specific naming convention or issue barcodes, but the program has been responsible for creating the number. See the previous section on Specimen IDs for a number of program OIDs (2.16.840.1.113883.3.5.8.4.1) being used to identify the issuer of the specimen IDs.

Program-issued referral numbers apply at the specific test level and are found in ORC.4 "Placer Group Number."

```
<ORC>
...
<ORC.4>
  <EI.1>SCMKM040915</EI.1>
  <EI.3>2.16.840.1.113883.3.5.8.5</EI.3>
  <EI.4>ISO</EI.4>
</ORC.4>
...
</ORC>
```

It is possible for a sample collected by and for one program to also be used as a convenience sample for another surveillance program. In that case an additional ORDER group would have a different referral number with a different Program OID.

Sometimes programs don't issue referral numbers. But the testing is still being performed on behalf of a specific program. In this case, the referral number is omitted and *only* the assigning authority is populated with the Program OID. This sounds like nonsense but is actually very faithful to the way identifiers are handled in newer versions of HL7 and with the way the components of the Entity Identifier (EI) data type are actually named in the standard. EI.1 is simply Identifier and EI.3 is Universal Identifier. Because OIDs are inherently universally unique, they can stand alone as unique IDs.

```
<ORC>
...
<ORC.4>
  <EI.3>2.16.840.1.113883.3.5.8.5</EI.3>
  <EI.4>ISO</EI.4>
</ORC.4>
...
</ORC>
```

So, when asked for the “Program OID,” make sure you know the context. Do you want the program that issued the specimen label or the program that wanted the test results? If you want information on the specimen label, look at the specimen identifier. If you want the program for the test, look in that test order's ORC.4.

### Observations

Values by themselves don't mean much. If I tell you that the answer is “<1:8” I haven't given you anything worth knowing. If I tell you that the influenza H7 antibody titer on the blood sample “A” from bird “1234” on farm “005AB6Q” drawn for investigation “SC05002” was “<1:8” then you know something. For the preceding to make sense in computer terms, each of the identifiers for tests, animals, farms, and cases must be clear, universally unique and unambiguous. The NAHLN uses identifiers extensively to provide context for the actual data. The preceding sections have explained how these identifiers are connected in the nesting structure of the message. We also need a way to keep the questions and answers straight.

In *The Hitchhiker's Guide to the Galaxy*, a pandimensional race of beings got so fed up with the constant bickering about the meaning of life that they built the ultimate super computer and set it to the task of finding the ultimate answer to the meaning of life, the universe, and everything. Seven and a half million years later the computer produced the answer: “Forty-two.” The problem was that in all these years, no one had *ever* known exactly what the question was. Such is the nature of computer answers. They have meaning only to the extent that we know what the question is.

One segment keeps popping up everywhere in the NAHLN Result Message. This is the Observation segment; OBX. The OBX segment holds among other things an attribute

value pair in OBX.3 and OBX.5 respectively. It keeps the questions and the answers together so we can avoid the “42” problem. Just about anything can be noted using an observation. The NAHLN/LMS messaging system uses OBX to make observations about the case, the patient, and the specimen, as well as the actual test results. Keeping each usage of OBX straight is essential. We gain this contextual information from both the location of the observation segment in the message and from the observation identifier it contains. The various ways OBX is used are listed below.

### Patient History

Relevant patient history can be included by populating the OBX segment in the Patient\_Observation group. This is not intended to provide a complete medical case history, but rather to convey information that is important to the conduct or interpretation of the ordered tests.

### Physical Findings

Similarly, specific lesions noted that may have informed the submission can be included by populating the OBX segment in the Patient\_Observation group. This is *not* where observations as part of testing are made. Instead, it represents observations about the patient made by the attending veterinarian, or by the lab that then added the NAHLN tests to an existing case. “I ordered the following test(s) because I saw ...”

### Additional Information About the Specimen

While the Specimen segment (SPM) includes specific fields for such information as the type, source, and condition, it is sometimes necessary to send additional information about the specimen.

### Results

Of course, every result message in the NAHLN messaging system will include at least one OBX in the Result group. This OBX will be the answer—or one of many answers—to the question in OBR.4, the universal service identifier. The specific observation type is in the observation identifier, OBX.3, and the value in OBX.5 and/or the interpretation in OBX.8. In day-to-day operation of the laboratory we often blur the line between actual test results and the laboratory interpretation of those results. In general, if an instrument produces a number or numbers, those are the actual results and will generally have a corresponding LOINC code with some quantitative type. In those cases, the laboratory interpretation of the test results positive, negative, intermediate, etc., go in the misnamed OBX.8, “Abnormal Flags” field. (This field has been renamed to “Interpretation” in HL7 version 2.7.) If the test results in only a qualitative color change, etc., then the actual result may be positive or negative.

Note: Laboratory interpretation is the laboratory classification of the individual test result based on invitro performance of the test during validation. This is distinct from the act of making a clinical diagnosis or classifying a case such as what a Brucellosis epidemiologist does when reviewing the sequence of tests performed by the laboratory. It is quite possible and even common for a “positive” test result to lead to a case classification of “suspect” or “presumptive.” Because case classification is often closely

tied to, or even algorithmically derived from, laboratory interpretation, it is easy to confuse these two types of test result interpretation.

### ***Reason for Study***

For tracking purposes, it is important to know whether testing is being performed as part of a routine surveillance program, in response to an outbreak investigation, or for some other reason. This information should be sent in each Observation Request in OBR.31 “Reason for Study.” Its value is a local code with values including “N” for national surveillance programs, “S” for surveillance and “F” for FAD diagnostics. So, what is the difference between “Surveillance” and “Surveillance, National?”

The reason “Surveillance, National” is the recommended default reason for submission for VS recognized national surveillance programs (when another specific reason is not provided). The following are considered recognized national surveillance programs: Classical Swine Fever, Bovine Spongiform Encephalopathy, and Avian Influenza - Wildlife Avian Surveillance. As other tests join the NAHLN messaging system, this list has grown. A good source for the full list is in the OID repository under the arc for AHER Programs.

On the other hand, the reason “Surveillance” is sometimes used for surveillance in buffer zones during outbreaks of non-FAD diseases or in areas where there is a need to provide evidence disease no longer exists. Uses outside the NAHLN proper will require additional codes.

Be sure to check any program guidance for the use of the Reason for Study codes supported. There is a growing tendency to assign more fine-grained precision based on how a specimen was selected for inclusion in a surveillance stream. The codes listed here are likely to be incomplete by the time you start messaging under a new program.

### ***Parents***

No, not *that*, obvious, kind of, “parent.” If the pedigree of an animal is important, that would go in an observation on the patient. In this section we will address situations in which one specimen is drawn from another specimen, rather than directly from the patient, as when a specimen is sub-sampled. This also applies when the opposite is done and a number of specimens are pooled into one for testing. We will also address the case where one order is in response to the results of an earlier observation.

Before we throw out the idea of a pedigree, let's think about what we mean by that. The first definition of “pedigree” is “A line of ancestors, a lineage.” We can generalize that as “an earlier similar thing that gave rise to this one.” When we use the concept of *parent* in a NAHLN message, we mean where something came from. A specimen that has a parent is a specimen that came from some other specimen. An observation request that was made in response to the results of an earlier test, such as a confirmatory test, has that earlier request as its parent.

Take a close look at the specimen segment SPM, specifically at SPM.2 and SPM.3. These fields are “Specimen ID” and “Specimen Parent IDs.” Each is the same data type, Entity Identifier Pair. The only difference is that SPM.3 can repeat. A specimen has one identity but can have many parents. The pair part comes from the fact that this one identity can have one identifier assigned by the submitter (placer) and another by the lab (filler), but both identify the same specimen. Each of these fields identifies the same kind of thing; a specimen. But each has unique meaning.

The specimen ID identifies the specimen actually tested in the tests in the observation request or requests in this Specimen group. Since each distinct specimen gets its own Specimen group, each SPM segment can have only a single specimen ID field with one or two identifiers. The placer identifier is any identification that came attached to the specimen. In some cases in the NAHLN this will be a bar code number. In an order message, only the placer ID will be populated since the submitter doesn't yet know what ID the lab will assign. The filler identifier is an identifier assigned by the laboratory—usually automatically by the LIMS system—for internal tracking. It is essential that both are kept associated with the results so that it can be identified at any point in the system. The placer ID will only be globally unique if assigned by an identified program using barcodes, etc. It may be just a simple label. But the filler ID should always be globally unique because it is assigned by the lab and scoped by the lab's OID in EI.3. The filler ID must always be provided and be globally unique.

The specimen parent IDs field can have zero, one, or many values. In each case the parent specimen IDs are copies of the SPM.2 field from their own SPM segment (even if this segment is not actually transmitted). If it has zero, this specimen has no parents. Don't feel sorry for it as an orphan. This just means that this is an original specimen not derived from any earlier identified specimen. If it has one identified parent, then this is a sub-sample such as an aliquot taken from a serum sample or a tissue sample taken from a submitted organ or larger tissue sample. If the specimen has two or more parent specimen ID's, it is said to be a pooled specimen. A specimen's pedigree can become very important in the chain of evidence created by confirmatory testing. By each specimen correctly identifying any parent specimens, this chain is kept intact. For practical purposes if a tested specimen is an aliquot of an aliquot, etc., and the intermediate specimen has not been included in the message, it is best to send the original specimen (maybe we should call it the “grandparent”) as the specimen parent ID so we have a record of the placer's original identifier.

Orders, or “Observation Requests” in our current terminology, can also have parents. In this case, the results of the test first ordered may result in additional tests being more-or-less automatically run. Examples include cultures that result in sensitivity only if a pathogen is grown, or serotyping only if certain viruses are identified. In NAHLN usage, this kind of parent-child relationship occurs in the laboratory, so the two are linked by the laboratory's (filler) order number from the initial test becoming the parent filler order number (OBR.29.2) in the additional test's OBR segment. This use case may become important as the NAHLN/LMS system is used for more referral confirmation testing. It is not used in any of the current applications.

## Case Numbers

Whose case is it anyway? An epidemiologic investigation may consist of a number of diagnostic procedures and laboratory submissions to numerous labs. Each laboratory has its own concept of a case or “accession.” The Foreign Animal Disease Diagnostician (FADD) or other investigator in the field needs to be able to link results to their case, and the lab needs to be able to keep its results organized by its accession number. All of this needs to flow through a national system and still make sense when it comes out the other end.

The NAHLN/LMS messaging system messages are laboratory centered. The “Patient Visit” in a NAHLN message is a laboratory case. This has been discussed under “Attending Doctor” (PV1.7) earlier. From this perspective, the “Visit Number” (PV1.19) is the laboratory accession number or case number. That leaves us needing a way to keep track of the epidemiologic case number.

Because foreign animal disease (FAD) cases are identified by a referral number assigned by the Area Veterinarian in Charge (AVIC) of the state in which the case occurs this number is assigned as part of the order. We find the Referral Number in the Common Order segment in a field that is rather vaguely named, “Placer Group Number.”

Veterinary Services Referral Numbers follow a convention that ensures that they are globally unique as long as we know exactly what they are. These numbers are identified by the OID “2.16.840.1.113883.3.5.8.x.” The terminal “x” represents a number specifying the program from which the number originated. For example, CSF surveillance numbers have a terminal “x” of 3, while AI surveillance has a terminal 4. Other number systems such as FAD referral numbers or FSIS condemnation numbers may have their own OIDs yet to be assigned. For foreign animal disease investigations, the VS referral number is the preferred identifier for the epidemiologic case. Other investigation numbering systems can be used as long as they are identified by a properly assigned OID.

## Coded Entries

The human mind is very good at figuring out the gist of ambiguous or “fuzzy” descriptions of things. Computers are generally much better when each input has a single deterministic meaning. We achieve reliable interpretation of the data by coding many of the concepts that in human-to-human speech would be carried by words. HL7 carries these coded concepts in a special type of field called a “coded entry.”

For the vast majority of your NAHLN/LMS messaging, you will be able to lookup exactly what should go in each of the coded elements.

Then all you need to do is map your local codes to those standard codes and build the appropriate CWE element. But there will be cases where you need more detail, alternate

## Reliability

Actually, people aren't as good as they think they are at consistently interpreting many of these “fuzzy” descriptions.

Researchers have studied the ability of words to evoke the same meaning when read by different clinicians (inter-rater reliability) and at different times by the same clinician, even the one who wrote them (intra-rater reliability). The results are not encouraging. Even 80% inter-rater reliability is seldom achieved. And some commonly used phrases have been proven to be virtually meaningless.

codes, etc. The next few paragraphs explain the additional features available in HL7 coded elements.

The most important components of a coded entry are the code, the human readable text for the concept, and a code for the system of codes used: “SCT” for SNOMED Clinical Terms, “LN” for LOINC codes, or “L” for local (NAHLN) code systems. In many cases our LIMS systems will have the data coded in some other coding system than the one required by the NAHLN. Mapping tables are used to look up the NAHLN-standard code. These mappings may be less than perfect one-to-one matches however, so we may want to send our original—or “alternate”—coding system. Three extra components are provided for this alternate system’s code, description, and coding system. We can also send the original text if this was coded from plain-old human-only words. Depending on the way the coded entry is used, it may be allowable to send only the alternate system, or even only the original text. This is determined by the coding strength.

Prior to version 2.5, HL7 put all coded entries in one data type amazingly called, “Coded Entry” (CE). Starting with version 2.5, two versions of CE were introduced to specify the coding strength. “Coded With Exceptions” (CWE) allows sending alternate codes or text without the preferred code. On the other hand, “Coded No Exceptions” (CNE) requires that the specified coding system be populated. If the concept in a CNE field cannot be coded in the required system, it isn’t worth trying to send. Coded No Exceptions is, therefore, generally used for structural fields or things where there is no sense in messaging anything non-standard. NAHLN test codes fall in this category because we only send tests for which we *know* LOINC codes exist.

Starting with version 2.6, all CE fields specify either CWE or CNE. You will still hear casual reference to “CE,” but in actual messages these are always one of the two specified forms. The two data types are identical in structure, differing only in the requirement to use the specified coding system in the primary coding system components.

The basic structure of the NAHLN Messages is also being used in less constrained applications such as between laboratories and from laboratories to state animal health authorities. These applications often need to use Coded With Exceptions (CWE) where the NAHLN uses Coded No Exceptions (CNE). In order for the more relaxed schema to work with the NAHLN schema--for the NAHLN schema to be a “proper constraint” of the relaxed one--the NAHLN has reverted to using CWE for tag names in both types of coded entries. The difference comes in the enforcement of the codes.<sup>19</sup>

### **Pooling**

*The Hitchhiker's Guide to the Galaxy* reminds us that things are not always what they seem. For example, mankind always thought he was smarter than dolphins because he had achieved things like the wheel, New York, wars and so on. Dolphins, on the other

---

<sup>19</sup> This is a consequence of the fact that the XML encoding rules for HL7 are secondary to the pipe-delimited encoding. In the pipe-delimited format, the only difference between CWE and CNE is the code strength so CNE is a proper constraint on CWE. In the HL7 XML encoding, the tag names reflect the datatype and so we cannot represent CNE as a constraint on CWE. It is a completely different datatype. So, don't worry about it, just use the <CWE.n> tags everywhere.



hand, hung around in pools doing back flips through rings and punching footballs with their noses. Unfortunately, mankind completely failed to recognize that the dolphins knew that the world was about to end and were trying to warn us. There was important information coming from those dolphin pools, and we failed to understand it.

There is important information in the results of pooled laboratory tests as well. But as with the dolphin messages, we need to be careful about how we interpret this information. Pooling involves combining two or more samples to form a composite specimen on which the test is actually run. This pooling saves time and money and can produce excellent results if properly designed and implemented. The test must be designed and validated for use with pooling. And the pooling must be performed in a manner consistent with the test validation. Some tests may be performed on pools of five—or some other number—of samples from the same species of animal.<sup>20</sup> Using properly validated methods, negative results on pooled samples give good confidence that all samples in the pool were negative.<sup>21</sup> On the other hand, positive results on the pool indicate that one or more samples in the pool are positive. We cannot assign a “positive” value to any individual sample, but only to the pool as a whole. Generally this result will be followed-up with testing on the individual samples—either at the testing laboratory or upon referral to NVSL.

The challenge in sending pooled testing results via the NAHLN messaging system is keeping all the individual samples and pools straight. We must be careful not to assign results to the wrong level of aggregation, yet we must provide enough information that the system can “know” which samples—and ultimately which animals—were represented by a pooled test. Three details require attention: demographics, sample identification, and results formatting.

### **Demographics (Signalment)**

In the NAHLN messaging system, the PATIENT group—consisting of a PID segment and possibly associated observations—is used to define not necessarily one individual but instead the demographics of the entity represented by this ACCESSION\_DETAIL group. This may be a single animal. Unlike human hospital messaging—but like much public health data—this may also be a herd, flock, or other grouping. One such grouping is formed when we pool specimens for more efficient testing.

The nature of the pooled testing requires that all samples that make up a pool come from similar animals. They will also have the same general address, etc. For purposes of the pooled testing, we are not sending individual animal identifiers, only individual specimen identifiers. Thus, we are able to use a single PATIENT group to identify the animals

---

<sup>20</sup> See specific method standard operating procedures for specific requirements for specimens going into a pool for any given test method.

<sup>21</sup> The test design must take into account the fact that we have pooled any interfering substances as well as any target RNA. If tested individually, the samples with inhibitors might be false negatives, but any other positive samples would be positive. When pooled, an inhibited sample may interfere with another sample that would have tested positive individually. It is also possible for pooling to dilute inhibitors and allow a positive to show up when it would have been negative in individual tests. But these are issues for those who design and validate the test methods for use with pooling. Pooled results have their own interpretation issues just as do individual tests.

represented in the pool. The value of each field in the PID segment should be the most specific value that is true of all animals in the pool. For example, if all animals are the same breed or subspecies, you could code the species field of PID to that level. If they are different subspecies, the most specific value that is true of all the animals would be the species. If we send mixed species, we need to go up the taxonomy tree to the lowest common designation, perhaps genus or family. (Few, if any, testing programs will permit pooled testing on pools consisting of different species however.)

### Identifiers

While we do not always send individual animal identifiers, the identity of the individual animals or their group is known by the submitters. Results reported with a sample identifier can be associated by the submitter with at least a group of animals, if not a specific animal. Thus, results for pools must report both the pooled sample identifier and the identifiers of all the samples that contributed to the pool. In HL7 these individual samples are identified as Specimen Parent IDs in SPM.3. SPM.3 is an Entity Identifier Pair field (like SPM.2) that repeats. So for a pool of five source specimens you will have one SPM.2 with identifiers for the pool and five <SPM.3> elements each with a placer identifier (the label on the tube) and a filler identifier (your system's way of identifying the individual samples submitted.) SPM.2 will not have a placer identifier because the pool was created at the lab. This is the only case in which the placer ID may be omitted. The submitter, when they get the results, will be able to pull out their placer identifiers from the Specimen Parent IDs and lookup the animals represented in this pool. When the individual parent IDs are sent, the number of samples in the pool sent in SPM.13 "Grouped Specimen Count" must agree with the number of parent specimen IDs sent in SPM.3.

For pools created by the submitter, the Specimen segment contains the identifiers for the submitted specimen exactly like any other specimen. The number of samples in the pool is sent in SPM.13 "Grouped Specimen Count."

In either type of pooled specimen, the fact it was pooled is indicated by the Specimen Role (SPM.11) to "L" for "Pooled". Wait, what? "L for pooled?" Well, "P" is used for "Patient" meaning it is the actual specimen drawn from the patient. At least there is an "L" in "pooled" but not in "patient".

### Other Specimen Details

Unfortunately, the message used in the NAHLN does not allow instances of the specimen segment other than the ones tested. This means that the rest of the specimen details in the specimen tested apply to *all* the specimens that make up the pool. This will be true in the vast majority of legitimate pooling situations.

### Flavors of NULL

In *The Hitchhiker's Guide to the Galaxy*, one of the consequences of Zaphod Beeblebrox's use of the infinite improbability drive on the spaceship *Heart of Gold* is that

an infinite number of monkeys show up outside Arthur's door with a copy of *Hamlet* that they have typed up. What does this have to do with the concept of "Null?" Nothing really; it is simply a gratuitous segue to Shakespeare and a play that has a lot to do with nothing. This is, of course, *Much Ado About Nothing*. In Shakespeare's day the words "nothing" and "noting" were pronounced the same. This could lead to any number of puns about *noting* this or that when it was really *nothing*. And, as it turns out, the NAHLN presents many situations in which it is essential that we take *note* of *nothing* and even send this information in the content of our messages. When we send nothing in a field we are said to be sending a *null*. When we send an explanation of why we are sending a null, we are said to be sending a *flavor of null*. Flavors of null turn out to be, "much ado about nothing."

When you ask a computer a question you may get an answer or you may get . . . nothing. The nothing you get has some very interesting features. Most data types have very specific logic for performing arithmetic and logic with these null values. Besides this computer logic, we sometimes need to know *why* we got a null instead of an answer.

Why would we end up sending nothing? First, we should note that for some fields it makes no sense to send nothing. Sending nothing in the test type field, for example, would make the message useless. In these cases, the message profile and business rules do not allow null values. In other cases however, the message structure requires fields for things like animal identifiers that we may or may not be able to send. Because the message structure requires *something* here, we need to send a placeholder. And, if we aren't sending an actual value, it may be useful to know whether we aren't sending it because of confidentiality rules or because we don't have the information. In the former case, it would be worth requesting the sender to follow-up on a positive. In the latter case, the sender doesn't have the identity to begin with. In other cases, our standard message structure may have permanent fields that are never relevant to the NAHLN and therefore hard-coded as "not applicable."

As part of the development of version 3, HL7 has defined a number of these flavors of null. The NAHLN has adopted some of these and added a few other NAHLN-specific codes that can be used to specify not only that the data are not provided, but why. When we cannot or do not provide a value for a required field, we populate the field with a specified code string describing the reason why the field is left null.

### HL7 Null Flavors

| Code | Name                    | Definition   |
|------|-------------------------|--|
| NI   | No Information          | No information whatsoever can be inferred from this exceptional value. This is the most general exceptional value. It is also the default exceptional value.   |
| OTH  | Other                   | The actual value is not an element in the value domain of a variable. (e.g., concept not provided by required code system).  |
| NINF | Negative infinity       | Negative infinity of numbers.  |
| PINF | Positive infinity       | Positive infinity of numbers.  |
| UNK  | Unknown                 | A proper value is applicable, but not known.   |
| ASKU | Asked but unknown       | Information was sought but not found (e.g., patient was asked but didn't know)   |
| NAV  | Temporarily unavailable | Information is not available at this time but it is expected that it will be available later.  |
| NASK | Not asked               | This information has not been sought (e.g., patient was not asked)   |
| TRC  | Trace                   | The content is greater than zero, but too small to be quantified.  |
| MSK  | Masked                  | There is information on this item available but it has not been provided by the sender due to security, privacy or other reasons. There may be an alternate mechanism for gaining access to this information. Note: using this null flavor does provide information that may be a breach of confidentiality, even though no detail data is provided. Its primary purpose is for those circumstances where it is necessary to inform the receiver that the information does exist without providing any detail. |
| NA   | Not applicable          | No proper value is applicable in this context (e.g., last menstrual period for a male).  |
| NP   | Not present             | Value is not present in a message. This is only defined in messages, never in application data! All values not present in the message must be replaced by the applicable default, or no-information (NI) as the default of all defaults.   |

| Code      | Name      | Definition  |
|-----------|-----------|---|
| NON HUMAN | Non Human | Used to indicate that the Surname is not being supplied in the HL7-required Family Name field because this is not a person. This serves to protect against misinterpretation of the data when messaging with public health or other agencies used to receiving only human data. |

In NAHLN usage, the common codes are MSK for any field supported by a message and where data are known by the sending lab but not sent for confidentiality purposes; and UNK for any field supported by the message but not known by the sending lab including when the information was not provided to the lab for confidentiality purposes. The distinction determines whether it is possible to seek the information from the lab in special circumstances that might warrant re-identification.

The phrase “NON HUMAN” has been used in other public health related version 2.x messaging to fill the Surname field with something that will flag records as not being from humans so they will not inadvertently be counted in human statistics. Because we need to put something in Surname to satisfy a general HL7 rule that might break a compliant parser, we follow this convention. This field may also be populated with, “Not Provided.”

Note: To date, the coding of these flavors of null has been very haphazard. If receiving messages your system needs to be prepared to receive either the code or a fully spelled-out string such as, “Not Applicable.”

## ***Sending Nothing***

Many implementers fall into a tempting trap by routinely sending empty tags everywhere they don't have values. This fails in several places where the element is optional but has a data type that does not allow null or blank values. It is almost always best to omit the entire element at the level below which no data are available. The few exceptions are structural elements when the HL7 standard requires that we send placeholder values. These are clearly spelled out in the schema and implementation guide.

Sometimes we need to be even more explicit. What if instead of saying nothing, we need to say, “The answer is nothing.” This tells the receiver that even if they think they know the answer, we are telling them to replace that with nothing. Sending null as discussed above is not this specific. In HL7 if we want to instruct the receiving system to replace existing values with nothing (also known as “delete”), we send a pair of quotation marks with nothing between them. In XML this ends up being a pair of tags containing just the two pairs of quotation marks (`<CWE.1>""</CWE.1>`).

## ***Lab-to-Lab Order Messaging***

When an order message comes from another laboratory rather than from the original submitter, a number of message elements become less clearly defined semantically. This section addresses the usage of the NAHLN Order Message in lab-to-lab scenarios.

### **Redirection of Specimens**

One of the functions of the NAHLN is to provide laboratories with support during surges when their own testing capacity is exceeded. In these cases, some specimens arriving at the laboratory may need to be redirected to another laboratory. Eventually submitters will be instructed to submit directly to the backup laboratories. But in this scenario, the laboratory has an order and one or more specimens for which it needs to create an order at another lab to which it will forward the specimens. This same scenario might be used by a laboratory that has received an order for a test it does not perform in-house and is sending on to a reference laboratory.

This process will most likely be greatly refined in the future. In the meantime, results are to be submitted by the performing laboratory exactly as they would have had the specimens come directly to them; with one exception. An additional ROL segment should be added with a role code of "LAB." This designates the referring laboratory. The fields in this ROL should be sufficient to identify the laboratory.

### **Confirmation Testing**

A different scenario exists when a laboratory performs testing on a specimen and then sends part of that specimen—an aliquot—on to another lab for confirmation testing. In this case, the specimen being sent is not the original but is derived from it. In order to keep this relationship straight, we need to track the specimen as well as its parent specimen.

Whether it is the remainder of the original specimen in its original container with an aliquot removed for local testing or an aliquot in a new container that is sent to the confirming lab, it is logically a new specimen. We have both a specimen ID field and a specimen parent IDs field. The scenarios seem complicated. If you take each scenario slowly you should be able to follow the logic and create a good chain of custody using these IDs.

If the original container is sent with the original ID, then that ID is both the Specimen Placer ID and the Specimen Parent IDs, Placer Specimen ID. The specimen is its own parent! The Specimen Parent IDs, Filler Specimen ID will be the internal ID assigned to the original specimen by the referring laboratory—and is likely the label on its retained aliquot.

If the specimen forwarded to the confirming lab has a new label, that label is used as the Specimen ID, Placer Specimen ID. The Specimen Parent IDs, Placer Specimen ID will still be the original ID from the field. The Specimen Parent IDs, Filler Specimen ID is once again the internal ID assigned by the original lab—and likely correlates with the label on the tube sent to the confirming lab. The take-home lesson here is that the

Specimen Parent IDs, Placer Specimen ID *always* identifies the identifier assigned by the field, program, etc., that submitted the original specimen. The Specimen ID, Placer Specimen ID *always* corresponds to the actual identification affixed to the specimen sent to the confirming laboratory.

Note: There are very strict rules that cover foreign animal disease investigations and duplicate testing at FADDL and NAHLN labs. Where duplicate specimens are required they *may not* be aliquots from one parent specimen and will have two entirely separate specimen identifiers.

## Compromises and Less-Than-Ideal Solutions

[Insert something cute from HHGG, “Something almost but not completely unlike tea” perhaps.]

### Cycle Thresholds

Many results are numerical values. Most of us are fairly familiar with numbers. “Two plus two makes four” is commonly used as an example of the obvious. Another obvious feature of numbers is their ordered nature. We all know that six is greater than two and eight is greater than six. Some numbers are whole numbers only. Others can include a fractional part. But they all follow basic mathematical principles like these. But then along comes something like cycle thresholds in lab results.

Real-time PCR results are recorded as the number of cycles needed to raise the indicator reaction above some defined threshold—known as, “Ct value.” It is actually the point on an interpolated curve where the curve crosses the line, thus the fractions. In general, the higher the number of cycles, the *less* analyte was present at the beginning of the test.<sup>22</sup> In some cases there is so little—or none—of the analyte present at the beginning that the cycler could—in theory—run forever without reaching the threshold. Thus, each RT-PCR test has defined the maximum number of cycles it will complete, after which the machine simply stops and records a negative result. But the output of the machine is a number (real number, includes decimals). How does a single number record something like the machine quitting after a certain number of cycles? Various systems throughout the computer age have used different mechanisms for such things. One of the instruments used by the NAHLN adopted the convention of reporting zero to mean a negative test based on reaching the maximum number of cycles. Because this is a commonly used machine, the NAHLN uses this same convention. So, for a test with a cycle maximum of 40, a value of 0.1 to 40.0 represents the number of cycles needed to reach the threshold level. But a value of 0 or 0.0 means the threshold was never reached.

Back to numbers. In real-life 40 is greater than 0. In cycle thresholds where lower numbers mean “more positive,” zero is the least positive value of all. So, the normal ordering of numbers does not apply unless we exclude the special value of zero. This will affect all sorts of mathematical analysis such as averages, statistical distributions and

---

<sup>22</sup> Or, the less like the intended target DNA sequence. The point being that the higher the Ct value, the *less* “positive” the test.

inequalities. In all these cases, an extra clause will be needed to treat zero as a special case.

The NAHLN can move beyond this potentially confusing convention by treating cycle thresholds as true numbers and/or inequalities (e.g. >40). This requires careful analysis of all the potential machine output, etc. as well as coordination with all program staff used to seeing zero for “negative.” This more correct usage also has variants. It is most correct to send as a structured numeric with “>” in the comparator (SN.1) component and “40” in the first number (SN.2) component. Be sure the OBX.2 datatype field matches the way you send this.

### ***Result Interpretation, AKA “Abnormal Flags”***

A central problem in clinical epidemiology is the lack of inter-rater reliability of the kinds of interpretation words that we all like to use. That is, if two clinicians read the same pathology report, they are more likely than not to interpret the pathologist's words in significantly different ways.<sup>23</sup> One way we have of trying to overcome this is the use of standardized vocabularies such as SNOMED. Standards such as HL7 are moving strongly away from simple lists of words and codes to the use of these standardized vocabularies. This change will not take place overnight, however.

One area where the old-style simple lists cause the NAHLN result message to be less than ideal is the “Abnormal Flags” field (OBX.8). This field was originally used just to “flag” numerical results as “normal”, “high”, etc. Over the years it has evolved to be much more of a general use “interpretation” field. As such, it really should have a well-defined ontology of possible values. But, because of the way it is defined in HL7, we have only a simple list of values. This will change if and when the NAHLN moves to HL7 version 2.7. But these things take time.

### ***Non-results and Qualified Results***

There are almost as many ways a result can fail as there are quantum states in the Heart of Gold's “Infinite Improbability Drive.” Most of these can be expressed unambiguously through use of fields in the specimen and observation segments.

When producing a human-readable reports we frequently combine multiple facts into one word or phrase. These are fast and efficient for the human reader. On the other hand, they require very specific work to parse back out into individual facts when stored that way in an information system. Computer analysis works much more efficiently with what computer scientists call “first normal form” data that have one and only one fact in each field. A significant and problematic example of this divide is in how we handle qualified results.

Version 2.8 of HL7, from which we have pre-adopted several use patterns, specifically allows OBX.8 (“Abnormal Flags” or “Interpretation” depending on version) to be used

---

<sup>23</sup> D.L. Sacket, et.al., Clinical Epidemiology; A Basic Science for Clinical Medicine 2nd Ed., Boston, Little Brown And Company, 1991, p 109.



for non-result results. “This field may also be used to convey an assessment of an observation where no legitimate result may be obtained. This includes laboratory assays that are rejected due to the presence of interfering substances, specimen toxicity or failure of quality control.” For example, if there is a quality control failure, OBX.8 would get the value “QCF”. OBX.11 status would be “X” and OBX.5 would be omitted or blank since no value is available. Similarly, cytotoxic or anticomplementary samples would have “TOX” or “AC” in OBX.8 and “X” and blank for OBX.11 and OBX.5.

Some non-results are more complicated. If there is an identifiable problem with the specimen—wrong site, poor condition, etc.—that information needs to be conveyed in the appropriate field(s) in the SPM segment.

The HL7 Result messages, including the NAHLN's OPU message, include detailed information about the specimen in addition to the actual result. So, as confusing as it is, we need to examine two distinct segments—SPM and OBX—to find what shows in the human-readable report as simply result. And we need several fields from each.

SPM.4 Specimen Type

SPM.6 Specimen Additives

Rarely SPM.8 Specimen Source Site (Usually pre-coordinated into SPM.4 by Terminology Services)

SPM.21 Specimen Reject Reason (used even when we “make due” with the specimen received.)

SPM.22 Specimen Quality

SPM.23 Specimen Appropriateness

SPM.24 Specimen Condition (This is the most useful for details.)

OBX.5 Observation Value

OBX.6 Units (as appropriate for value)

OBX.8 Interpretation Code

OBX.11 Observation Status

To take an example we haven't seen yet but could, imagine a swab specimen submitted in the wrong transport medium (blood in the wrong anticoagulant, etc.). Maybe we can still run the test but aren't as confident in the result. The specimen type is the swab type, the additive is what it really came in. The specimen reject reason is “IA” for inappropriate additive, and the specimen appropriateness is “A” acceptable but not preferred. The OBX segment is the same as if the medium had been right. If we couldn't run the test, the SPM would be the same except specimen appropriateness would be “I” inappropriate. The observation result value and interpretation would be blank and the status would be “X” for no result available.

For a hemolyzed or autolyzed specimen the pattern is similar. The specimen reject reason is “SC” for specimen condition and the condition code is “HEM” or “AUT” respectively. The specimen quality code would be “P” or “F” for poor or fair depending on the extent. The specimen condition code is of a type that allows us to add codes in the future if the supplied list is missing anything we need.

For specimens that are of the wrong type or from the wrong location, the same pattern applies. The actual specimen type and sometimes location are sent in those fields. (The terminology services at Virginia Tech have created pre-coordinated specimen types for the most common such as “Brainstem specimen not at the obex” so we seldom need to separately populate the source site.) The reject reason is “IS” inappropriate specimen and the specimen appropriateness is either “A” acceptable but not preferred if the specimen is useable at all or “I” inappropriate if not testable.

The NAHLN terminology has “ILF” (Insufficient Lymphoid Follicles) as an Interpretation value but in most of the cases in Scrapie the reason for insufficient follicles is a problem with the specimen. OBX.8 in these cases will contain “ILF” but it is important that the specimen segment give as much information as known about why, wrong site, etc.

### ***Test Interpretation vs. Case Classification***

The Abnormal Flags field is used to convey the interpretation of the laboratory test output. It interprets the *invitro* results. A large number of additional factors are needed in order to translate these lab results into a clinical or epidemiological interpretation of the case. Some tests that have high sensitivity but low specificity are used as screening tests. In these cases a positive lab result never leads directly to a positive case classification but only triggers further testing. The case classification would likely be called “potential positive.” In other cases positive lab tests are routinely confirmed by national laboratories. The case classification in these cases is often something like “presumptive positive” or “suspect positive.” Depending on the current prevalence of the disease in question, more or less action might be taken based on the initial laboratory results. If the prevalence is high enough the confirmation step might even be dropped and cases classified based on initial testing. It is essential that we keep the concepts of laboratory test result and case classification clear.

The NAHLN result message includes both the raw and interpreted laboratory results. These are assigned independently from the epidemiologic factors that affect case classification. Case classification is generally performed by clinicians or regulatory officials, *not* by the laboratory. Labs often *know* from experience what case classification will result from a specific set of lab test results but they do not actually assign that classification. The NAHLN result message does not include an element for the case classification. Part of the confusion about this comes from the fact that similar words are used in both test results interpretation and case classification. There is a fear that the word “Positive” in the context of test result interpretation will be misconstrued as a “Positive” case classification.

It *may* be necessary to use case classification synonyms for the Abnormal Flags field in some rare cases. These would be cases where misinformed reading of the raw messages could lead to economic or political consequences. If and when this is done, it must be clearly noted in the program guides and the broken semantics taken into account in all subsequent analysis.

## Repeated Testing or Re-Sent Results

Sometimes the same test shows up for the same accession and same specimen. How can that be? Some different scenarios might apply. First, the system might simply be sending the same information over again. Perhaps its log of sent results is missing something. If the receiving system gets the same final result for the same test, for the same specimen in the same accession, it will assume this to be the case. (A final result is one with an "F" in OBX.11 Result Status.) If another message with a different result also marked final comes in, the receiving system should assume an error has occurred.

Because our data systems are less sophisticated than those on the Heart of Gold, this may need to result in the accession being flagged for human intervention, etc. If the sending system intends for the changed result to correct the previously sent final result, it must flag it as a correction by putting "C" in the OBX.11 field. An observation with status "C" and the same value would be odd but is not an error.

Repeating a result with "F" status is allowable as long as the value and interpretation are unchanged. This is a common practice when sending additional results on a previously messaged accession. Some systems send only the new results while others, for simplicity, send the complete accession each time.

But what if the lab actually ran the same test over again on the same specimen and *both* results should be submitted? This is the job of the Observation Instance Identifier (OBX.21). Its purpose is to uniquely identify observations even when all the associated identifiers are identical. This field is an Entity Identifier so it can hold the LIMS system's internal result identifier along with the lab's OID, (and "ISO" to designation the OID as an OID). A second repeat of the same test on the same specimen can be easily differentiated by a different OBX.21.

**Action by Receiving System**

| Original Status | New Status | Values            | Instance ID   | Action       |
|-----------------|------------|-------------------|---------------|--------------|
| F               | F          | Same              | Blank or same | Ignore       |
| F               | F          | Different         | Blank or same | Signal Error |
| F               | C          | Different         | Blank or same | Replace      |
| F               | C          | Same              | Blank or same | Ignore       |
| F               | F          | Same or Different | Different     | Add New      |

*NOTE: Before schema release 2.0 the C value was not yet supported. Be sure to check your current schema before implementing correction messages.*

## Changes to "Orders"

Sometimes an accession gets messaged and later additional information about the Premises, Owner, Animal, etc., comes into the lab and gets entered in the LIMS. These cases should be relatively rare. However, there has been a desire to send these changes in messages. Ordinarily in a typical HL7 environment like a hospital, these would be sent by the admissions department as an Admission Discharge or Transfer message of some

type. Because the NAHLN handles lab results alone, we need a way to send these in our laboratory result message. The conceptual approach is to send a message with instructions to replace the order.

In normal usage there are two instructions coded in ORC.1 “Order Control Code.” The message may create a new order, “NW” or indicate a status change “SC” of an existing order such as when additional results become available. Once order messages are widely implemented the results will have an order control code of “NW” only when the lab is originating the order rather than responding to an electronic order message—that indicates the order exists.

If we think of the whole message except the result as the “order” then if we want to change anything except the results, we need to replace the order. The order control code for this is “RO” for Replace Order.

If this makes sense to you, you may be wondering why this is in the confusing topics section. Well, enter ORC.5 “Order Status”. The NAHLN supports two values for this code in the results message. “CM” means that the order is complete. This message contains either all the results or all of the remaining results if partial results were reported earlier. “A” means that some but not all the results are complete. There is a not yet supported code “RP” that says that the order has been replaced. If the full order, order-response, result message sequence was in place, you might receive an “RO” in an order message from the submitter or LMS and need to reply with a response message. That message would have “RP” in ORC.5.

So, to make a tricky subject simple: “RO” in ORC.1 means “Please replace the information you have for this accession with the contents of this message.” If we supported it, “RP” in ORC.5 would mean, “OK, I've replaced the information for that accession.”

## NAHLN Standard terminology

“The Babel fish,” *The Hitchhiker's Guide to the Galaxy* says, “feeds on brainwave energy received not from its own carrier but from those around it. It absorbs all the unconscious mental frequencies from this brainwave energy to nourish itself with. It then excretes into the mind of its carrier a telepathic matrix formed by combining the conscious thought frequencies with nerve signals picked up from the speech centers of the brain which has supplied them. The practical upshot of all this is that if you stick a Babel fish in your ear you can instantly understand anything said to you in any form of language. The speech patterns you actually hear decode the brainwave matrix which has been fed into your mind by your Babel fish.”

If the Babel fish were more than a fictitious creation of Douglas Adams' imagination, we would not need standard terminology and this chapter would be finished. In the real world, however, it is up to us to perform this universal translator function. There are very valid reasons why every LIMS does not use precisely the same terminology in either its user interface or its internal data structures. But for data originating in one LIMS to

make sense in the context of another—or a central database—they must undergo a process something like that performed by the Babel fish. There are two options for performing this function. We could follow a pattern almost precisely the same as the Babel fish and translate each sender's native terminology directly into each receiver's version. This would result in one translation for every sender-receiver pair. For AVMA's 73,000 member veterinarians to communicate a limited vocabulary of only 6000 concepts would require over one hundred ninety thousand trillion translations. Even by federal budget standards this is a big number. It is *much* more efficient to have each sending system translate from its internal vocabulary to a standard and each receiving system translate from the standard to its own lingo. This results in half a billion times fewer translations.

So how does one go about this translation? Much like the Babel fish, the first step is to understand the *meaning* of the source terminology. This sounds obvious, but it is perhaps the most critical and often overlooked part of developing a system. Words and codes in LIMS and other information systems often develop local usages that are not apparent from simple examination of the codes or words themselves. If these meanings are unambiguous in actual usage, direct translation will be possible. In many systems, the meanings have evolved to be context-specific. If the system can provide the necessary context then it is still possible to develop translations from context+term to standard terminology. The sections that follow in this chapter will provide more details about each type of concept. This information should help you find the correct standard concept code in each of the terminology systems used by the NAHLN.

On the standard terminology side of the equation, the coded concepts must be completely unambiguous even in the absence of context. Developing these standard terminologies is a science of its own and we have people who do that for a living. (Better than us.) Armed with a clear understanding of the source terms' meanings, one can use various tools discussed below to find the correct standard terminology system and concept code. Conversely, when presented with a concept code in a given terminology system, the tools can help the user find the precise meaning as well as additional information such as the broader class of which this term is a refinement. If the standard code is 26105007 "Holstein-Friesian cattle breed (organism)" and my system does not have a code for this, the standard terminology can tell me that this is a refinement of 385474004 "*Bos taurus taurus* subspecies domestic European cattle (organism)" which my system does have and is specific enough for my needs.

The NAHLN provides a service to help with finding just the right coded term for use in any of the fields in the NAHLN messages. This service consists of a web application in which you can browse for terms as well as a group of experts that can help out in a pinch. You still have to do the local part of the mapping; understanding what your local terms really mean in their local context, but the terminology service will help with the standard terminology side. Details about how to use this service follow at the end of this chapter.

## **LOINC**

The NAHLN uses Logical Observation Identifiers Names and Codes (LOINC) codes to unambiguously name assays at both the order and observation levels. An order may result in a number of observations, but each observation is atomic. Each result has a single value. Orders are coded in the Universal Service Identifier in OBR.4 and the observations are located in the Observation Identifier field OBX.3. The NAHLN needs very unambiguous and non-duplicative codes for these fields in order to produce meaningful aggregate analysis of analyses performed on a regional or national scale with input from multiple laboratories. And that is a major reason for the NAHLN's existence, after all.

So, what, exactly, does a LOINC code define? We've used the term "test" rather loosely up to this point. When we get down to the level of standard terminology we need to tighten up these definitions. (Test is often used to identify the order or the result or the method. For example: "run a glucose test", "the serum glucose test result is 114mg/dl", "the urine dipstick is positive for glucose". Because the term test is so ambiguous, we use the more specific terms "order", "observation or result" and "method".

While I can order a "chemistry panel," I cannot assign a result to it. I can only assign result values to the atomic components of the panel such as glucose, BUN, Albumin, or Chloride. Is the "disease X antibody titer" at one laboratory the same "test" as the "X AB TITR" at another lab? Obviously, besides knowing what disease X is, you need more information to answer this question. You also need a lot of technical expertise in the virology and laboratory methods involved. Assume for the sake of argument that you are an expert in these procedures and that you have both laboratories' operations manuals in front of you. Even then, how do you decide whether any differences are enough to need different codes? The LOINC manual is full of discussion around just this issue. There is no simple formula. As a good, general rule of thumb you can determine that if normal ranges differ significantly between two methods, or if the difference would be mentioned in routine conversation between laboratorians or between laboratory diagnosticians and their clinician clients, then the two assays need different codes. Minor differences such as brands of manufactured kits or details about sample collection are not differentiated in the LOINC code. (If important, there are other places in the message that we can communicate this information.) An examination of the details used to create a LOINC code may be informative about the level of detail needed to differentiate between "tests."

A LOINC code captures six aspects of an observation type: Component (analyte), Property, Timing, System (specimen), Scale, and Method. "Say, what?" Why would you ever care about such technical minutiae? Local jargon used to describe assays from a given laboratory's catalog is more than adequate to describe exactly one analysis. Such terminology is easy for the laboratory staff and clients to use and is perfectly appropriate in that setting. When trying to communicate with a global system, however, such local terminologies create collisions (same name used for different assays in different locales) and duplication (one assay with different names in different locales). Converting (mapping) local terminology to a universal coding system avoids these problems. To be

confident that the mapping is good enough, one needs to be able to examine the factors that make up this universal coding system.

If you end up working with an informaticist to LOINC code a new assay (or an assay new to the NAHLN), you can expect them to ask a lot of questions to get an idea of the importance of each of the six parts of the LOINC code. The descriptions that follow will give you some idea of what they will be asking and why.

### **Component**

What is being measured in this test? This seems like a very simple question but is really fairly tricky to answer correctly. If this is an antibody, is it specific to a given serotype or group of serotypes or is it common to all types of this species of organism? Is it measuring IgM or IgG? If it is a PCR, what can we really say is detected by the primers and probes used in this assay? To provide the right information in the coded value, these details must be expressed with enough, but not too much, specificity. So, some descriptions of components are, “Influenza A Virus Hemagglutinin H5 Antibody” or, “Classical Swine Fever Virus RNA.”

### **Property**

This attribute of an assay tells what information about the component you are measuring. Is this a titer for antibody, or just presence of RNA? Is it a concentration or a simple count? Looking at how the raw results are reported is often a good clue to figuring out this part of the LOINC code, as are any units attached to the value. If the units are milligrams per deciliter for example, you can conclude that you are measuring a “mass concentration” of whatever the component is. If the value is “<1:8” it is a pretty good guess that this is a titer, but you’d want to be sure it isn’t a fraction or ratio.

### **Timing**

For purposes of the NAHLN this one is easy. For all current and probably future assays, the measurement is being made at a point in time. This is the time when the sample was taken. Even if the assay takes months to run—like Johnes cultures—the physiologically significant point in time is when the sample was collected. That is when the agent was present or absent from the animal or environment. For other assays such as tolerance tests, metabolic studies, or physiologic monitoring, the results relate to an interval of time. But not for the NAHLN, so forget we even mentioned timing.

### **System**

The system is sort of another name for “specimen.” Somewhat like component, there are many degrees of precision that could be included. Do we say “serum” or “venous serum collected from the left jugular vein?” This is a professional judgment call, ultimately validated by the LOINC committee. In general, unless there are significant differences between options in terms of conduct of the assay or interpretation of results, it is best to group them, so you will almost always see “serum” in the contrived example above. On the other hand, for things like blood gasses, just “blood” might not provide adequate definition and, “arterial blood” might be a better choice. Keep in mind that our system will have detailed information about the specimen in the SPM segment. The system part

of the LOINC code is to tell one assay from another, not to differentiate the specimen types in assays that are otherwise the same.

### **Scale**

This item relates directly to the way the results are recorded. Are they from an ordered list (ordinal), from a coded vocabulary (nominal), or free text (narrative)? If the results are true numbers, the scale is quantitative. Don't jump to the conclusion that because the results are made up of digits that this is quantitative. SNOMED codes are long integers but definitely not quantitative. They are nominal. A somewhat oversimplified tool for deciding if a value is numeric is to ask whether doing arithmetic on it makes sense. If not, it isn't a "number." Does it make sense to ask you what twice your phone number is?

### **Method**

There is a good reason why the method by which the assay is performed is left for last. It can be tough when a new code is needed, at what level to specify the method, if at all. In the words of the LOINC manual, "Methods need only be expressed as part of the name when they provide a distinction between assays that measure the same component (analyte) but which have different clinical significance or have a different clinical reference ranges." As a general rule chemistries, counts, etc., do not need to specify method while serology, molecular genetics, etc., do. Even then, immune fluorescence is only specified to the level of immune fluorescence (IF) rather than to direct fluorescent antibody (DFA) or anticomplement immunofluorescence (ACIF).

Now finally, the good news. Most of the time, you won't have to deal with this six-headed monster. The assays that will be part of the NAHLN messaging system have been standardized. Part of that process is LOINC coding by experts. In these cases, LOINC coding is as simple as looking up in the Terminology Services that the LOINC code for procedure X is Y. Mapping to your local test catalog in these cases involves more a question of asking if your procedures match the NAHLN standard than just LOINC coding. But the same six questions probably come out in this process.

LOINC lets us name orders and results. But that is only a couple of fields in the whole message. What about more general vocabulary such as patient species, etiologic agents, and diseases? For these we need a more generalized vocabulary.

### **SNOMED**

Standardized Nomenclature of Medicine (SNOMED) is a general medical and veterinary vocabulary developed over the past forty years. It has drawn content from various sources such as older veterinary vocabularies like the Standardized Nomenclature of Veterinary Diseases and Operations (SNVDO) and Standardized Nomenclature of Veterinary Medicine (SNOVET), and medical vocabularies like the British READ codes.

SNOMED is much more than just a list of medical words, or even a list of medical words with their definitions. The real value comes in the relationships between concepts that are captured in the way SNOMED concepts are modeled. As a user of the NAHLN, you



don't have to become an expert in SNOMED modeling. That is why we have terminologists. But a basic understanding of what those terminologists are up to will make it easier to understand why they are as worried as they are about helping you get just the right concept code.

### **Advantages**

For very small vocabularies it may be enough to simply list the words that make up the vocabulary along with their definitions. We will see examples of some of these specialized vocabularies in the sections on HL7-defined and NAHLN-defined vocabularies. Once a vocabulary expands beyond a few tens of terms however, it becomes very difficult to perform any kind of search or analysis logic on simple lists. As vocabularies grow, they are better handled in a hierarchical structure. The Dewey Decimal System is an example familiar to those of us old enough to remember trying to find paper-based books in a physical library before the entire contents of the Library of the Pan-Galactic Congress could fit into a flash-drive.

As the list of concepts in a vocabulary becomes even larger, a single hierarchy is not enough to capture all the ways we may need to view the underlying knowledge. Libraries add card catalogs to allow classification of books on several different hierarchies such as title, author and subject. SNOMED, as a vocabulary of all of medicine, is one very large and complicated vocabulary indeed.

To tame such a beast requires a carefully designed multiple hierarchical model that captures the relationships between concepts. For instance, you might need to find a concept code for a specific microorganism. You *could* do that from a long simple list of scientific names of organisms. But what if you also wanted to search for closely related organisms? With a hierarchy of concepts based on the "is a" relationship you could find a larger classification such as the family and search all organisms that had an "is a" relationship to that family. For this, you would be using a single hierarchy. Now, let's say you wanted to do an analysis of all conditions caused by any of these organisms. You could then use the "has etiology" relationship to the organisms in your "is a" list and get all conditions caused by any of this family of organisms. You are then using a multiple hierarchy.

### **Modeling**

So, when you call up your friendly terminologist and ask for a code for, "familial snickerdoodlepoodle third eyelid palsy" they are likely to take a while to check on any concept codes that come up. They will make sure the relationships in the terminology properly define the concept, not just that they match the words. Later, as the details of "familial snickerdoodlepoodle third eyelid palsy" are worked out by the scientists who specialize in such things, the knowledge they gain needs to be reflected in the relationships in SNOMED. Sometimes this leaves concept codes that have incorrect relationships. These may be retired and replaced by more correctly modeled concepts. Your terminologist wants to make sure you get the correctly modeled concept, not just any word that says the right thing. And sometimes they are going to change their mind based on new knowledge or improved modeling.

### Corrections

When faced with a concept code that is in use in the NAHLN and that is found to be incorrect or obsolete, great care is needed to ensure that the database contains the correct information as originally submitted, but also has the most correct modeling available for analysis. There are two general situations. The first we will call “deterministic substitution.” This is when the terminologist can be certain that every occurrence of code X means Y. In these cases she may send out a list and say, “For every occurrence of the codes in column A, from now on use the corresponding code in column B, and for analysis perform the same substitution on existing data.”

If only they were all that easy. A significant reason for retiring concepts is that they turn out to be ambiguous. In some cases they mean one thing and in other cases something else. If these are discovered in existing NAHLN usage, the instructions for handling will be more complicated. We might call this “ambiguous substitution.” The terminologist will have to include the multiple new concepts along with the logic needed to decide which concept code applies in which cases. If we had complete knowledge of all our cases, we could then apply this logic and assign new codes. This won't always be possible for existing data. We can, however, look at our ongoing usage and make sure the correct codes are used in each new case to remove the ambiguity.

### Usage in the NAHLN

So, in which cases does it make sense to subject ourselves to this complexity? The overly obvious example is, “when what we are describing is complex.” SNOMED is used for things like patient species (“taxonomic classification”), diseases, etiologic agents, specimen types, anatomy, and others.

### Mapping

Now that you are ready to give up and go back to the Dewey Decimal System, let's look at how you can actually do SNOMED mapping without a PhD in terminology. Don't under any circumstances attempt to code straight out of a list of SNOMED concepts and descriptions. This is the only thing in the universe worse than having Vogon poetry read to you. In NAHLN usage we have a distinct advantage in that each field in the message has a distinct usage and therefore a strictly limited subset of SNOMED that makes any sense at all. So, if you were testing a *Strongylura exilis* (California Needlefish) and accidentally coded a 129249002 (Needle biopsy) that wouldn't make sense, even if the test on the Needlefish involved a biopsy.<sup>24</sup> This is a silly example, but you can see the value in a restricted list of concepts that *do* make sense for each field. There are many cases in SNOMED where the “words” sound exactly right but make no more sense than doing a “Needlefish biopsy.” The list of possible codes can be further restricted for the NAHLN because we deal with a limited set of tests, species, specimens, etc. Large, yes, but nevertheless restricted and nowhere near Vogon poetry level. These limited lists are available from the NAHLN Terminology Service (<http://vtsl.vetmed.vt.edu/nahln/main.cfm>).

---

<sup>24</sup> The author's college Russian language teacher admitted to having once translated a medical document such that it said the patient had cancer of the punctuation mark “colon” rather than the organ “colon.” This kind of confusion is not just a hypothetical problem.

So, you have a limited list of concepts in front of you and you don't see the specific words from your record system. The next step is to be sure you understand what the words in your record system actually *mean*! If you aren't both an expert on the laboratory tests and an expert on the coding of the LIMS system, you will need to get help with this. Especially as systems age, the terms used in them can take on non-intuitive meanings necessary to make the system work. If the users of the system are used to these non-standard usages, they may work fine. But they won't message correctly unless mapped to the meaning and not the words.

Now go back to the list with meaning in hand. Do any of the concepts match the meaning? If so, you have a match. If not, is there a more general concept that captures all the information needed at the NAHLN level? If your local description is "eyelid tissue from third eyelid of a snickerdoodlepoodle collected with a Smith-Jacobson needle," you can do some generalization. First of all, we know it is from a snickerdoodlepoodle because that is in the PID segment. But we can go further. Does it matter to the NAHLN that a specific type of needle was used? It might, so don't dismiss this out of hand, but often it won't. At this point you can look for "third eyelid tissue specimen" which you would likely find if "third eyelid palsy" (or scrapie) were a NAHLN disease. (It is 8011000009104 by the way.) And if you are just looking for virus in various tissues, you may not even need the third eyelid part, but that would take some analysis to be sure important information isn't lost by sending the concept as simply, "tissue specimen."

### Dealing With "Species"

Organism taxonomy is one particularly challenging area for SNOMED coding. There are several reasons for this. The most common problem is that very few of us, even scientists, routinely use strict Linnaean taxonomic names in routine speech or writing. We use common names. Common names can refer to a single species or an entire genus or family. They can also refer to some members of a genus or family, but not all. Once in a while a common term even refers to a collection of relatively unrelated species, members of multiple genera or families, that just happen to share some common trait. Sometimes, Linnaean taxonomy as recorded in the scientific names turns out to be incorrect. This is especially common in microbiology. In these cases, species eventually receive new scientific names. Finally, SNOMED itself does not yet completely capture the complete hierarchy of all commonly encountered species. Thanks to the NAHLN Terminology Services, SNOMED is getting much better in this area and now even includes a good percentage of common names. But it is not, and will never be, perfect. So, how can we ever get species mapped correctly?

As with the general discussion above, the first step is to make sure you know what *you* mean by the species term in your local usage. Find the Latin genus and species name. Google is a good tool for this. Just search for the common name and "taxonomy." Then pick a reasonably reliable source from the returned links. Read carefully to make sure you haven't found a "sound alike." Did you find a single species? If not, it may be that your species is really a whole genus or even family. That isn't a problem as long as it

correctly captures your real-world usage. The “species” field in the message should really be called “taxonomic classification” and is used that way in the NAHLN. If something other than species, make a note of the taxonomic rank (family, order, etc.). It is a good idea to look up and down the “family tree” a little to be sure your usage isn’t actually higher or lower in the tree. A good resource for this is the USDA’s Integrated Taxonomic Information System (ITIS) (<http://www.itis.gov>), currently the best listing of animal taxonomy. Sometimes you will discover that the common term you use actually refers to multiple species either within a single genus/family (but not the ENTIRE genus or family) or across multiple genera<sup>25</sup> or families. If this is the case, then it is impossible for you to choose a single Linnaean term to map to your common term. But that is OK too. Although Linnaean taxonomic classification is preferred, when that is not possible, common taxonomic classification can be used.

Now go back to the NAHLN Terminology service subset listing for species. Armed with the scientific name and taxonomy of your species or the common name if there was no single Linnaean classification, can you find it or a slightly more general listing? If not, this is time to call on the experts. There are enough real challenges in mapping species that you really don’t want to attempt mapping from the raw SNOMED tables by yourself. Besides, even if you did find a concept in SNOMED that wasn’t in the NAHLN subset on the Terminology service website, the NAHLN server validates messages based on the NAHLN subsets so a message sent using that concept would fail.

### **Dealing With Specimen Types and Source Sites**

As with taxonomy, specimen types can be described in various levels of detail. Just how much detail is required in a NAHLN message mapping is up to interpretation as is where to draw the line between specimen type (SPM.4) and specimen source site (SPM.8). The terminology service tries to maintain specimen types for any specimens that are commonly used. This means that it should be only the unusual cases that require any “post-coordination” using specimen source site. It is highly recommended that labs consult with the terminologist at the terminology services before resorting to the use of specimen source site. But source site is there when needed to add detail.

### **Getting Help**

By this point you should have the vast majority of your terms well mapped. What do you do with the rest? First, go back over the steps you used before. Do you really understand your local usage? Is this usage really unambiguous? Does it always mean what it did the time before? Many terms don’t code because they don’t really make sense except in each individual usage. If you have ambiguous terms, you will need to find a way to disambiguate them. This may involve bringing in additional information from other fields, etc. Once you have an unambiguous concept and are sure you understand it, make a list of the best matches you could find and why they aren’t sufficient. Now, you are ready to submit a request for help on the NAHLN Terminology Service.

In many cases, the terminologist may be able to help you find a concept that does match the supplied descriptions based on its modeling but that wasn’t obvious. This is

---

<sup>25</sup> The correct plural is "genera" but that doesn't seem to make the point as well as the misspelled version.

commonly the case when human medicine uses a different common name for the concept than veterinary medicine does. If your concept does not exist in the NAHLN subsets, the terminologist may need to go to the full database to find it. (Terminologists get special training to withstand the rigors of this; but sadly, not to withstand Vagon poetry.) If they find it there, it will be added to the appropriate field-specific subset and you can map to it. If not, the terminology service may need to create a new concept for NAHLN usage. This involves understanding the relationships described earlier in this chapter and constructing and connecting all the necessary hierarchies. The Terminology Service has a special section (extension) of SNOMED where they can build these new concepts and assign codes.

### ***HL7 defined terminology***

Some concepts are from very short, simple lists. Others are only meaningful as defining parts of the HL7 message itself. Many of these little, specialized terminologies are defined directly by HL7 in the form of HL7 tables. These tables can be found in the HL7 standard and are reproduced in whole or as a subset in the Implementation Guide.

Mapping to HL7 tables is usually simple. In fact, the values are often fixed as a structural part of the message. For example, the Message Structure field for NAHLN Results Messages is always "OPU\_R25." These values don't usually make much sense outside of the HL7 message so there may not be any equivalent in the LIMS. For those that vary such as the processing ID, the value is based on the usage of the message in a specific case (production, testing, etc.) and is determined by the system that creates the message.

In a few cases, HL7 tables get complex enough to require specific subsets and mapping for NAHLN use. An example of this is the production class of an animal, herd, or flock. These code sets will be supported by the NAHLN Terminology Service the same way as SNOMED and LOINC.

### ***NAHLN Defined Terminology***

The designers of the NAHLN messaging system tried wherever possible to use message structures and terminology from the broadest possible community. Thus, the general structure of the results message is nearly identical to that used by human laboratories to report results as part of the Laboratory Response Network (LRN). The LOINC codes used to name laboratory tests are the same as those used in clinical laboratories both human and veterinary. And the SNOMED codes for species are the same as those used by the USGS Fish and Wildlife Service.<sup>26</sup> There are however, some concepts that are unique to the NAHLN or at least to regulatory veterinary medicine as conducted at the USDA level. For these concepts NAHLN has developed local coding systems. These are almost exclusively very short lists of things like roles. They are generally handled by assigning codes to the tables defined by HL7 as user modifiable.

---

<sup>26</sup> USDA's Wildlife Services continues to use four character species codes defined by the American Ornithological Union (AOU) that represent mnemonics for the common names of some bird species. These codes must be mapped to SNOMED.

Examples of NAHLN defined codes include the various roles played by submitters, collectors, etc., and types of animal identification tags. These are small enough lists that they can be included in their entirety in the *Implementation Guide*. Mapping is generally very straightforward for these small code sets. Where there is any uncertainty in the usage of various codes, they are explained in the *Implementation Guide*.

### **Terminology Updates**

As mentioned above, terminology is not a static beast—new species are discovered, new tests are developed, terminologists change their minds.... This means that new terms WILL be added to the subsets, and some concepts WILL be retired or removed from the subsets. How can we all possibly keep up? More technology! The terminologists at Virginia Tech have developed a web service to synchronize the data between the Terminology Service and the NAHLN data repository servers.

How does this work? Each time a concept is requested by a lab or the terminologists retire a concept, the subsets at the Terminology Service are changed. Each day, the LMS makes a call to the web service which returns all concepts that have been added and all concepts that have been retired (along with potential replacements for the retired concepts) in XML format. The XML files are processed by the server updating the subsets used to validate messages. LMS will then forward a copy of the terminology update XML files via email to all labs registered with email addresses on the Terminology Service forum (You must specifically ask on the forum to be added to the Terminology Update list). Schemas for validating these XML files can be found in the Current Messaging Documents thread on the NAHLN Documents forum.

So, what do YOU do? First, make sure that the people responsible for keeping your mappings up to date are registered with current email addresses on the Terminology Service forum, and that they have requested to be notified of Terminology Updates. Then, wait for the updates to happen and the email messages with the XML files to come your way.

There are two XML file types you may receive. The first, and easiest to handle, is the Added Concepts file. This file contains all concepts that have been added within a given time period. (The timeframe in question is given within the XML.) If you are an XML aficionado, by all means extract the added concepts with whatever tool you enjoy. Or, use your favorite XML editor (or Internet Explorer will do) to open the file so that you can view the concepts that have been added. If you have concepts in your internal lists that you could not map, you may find the correct term listed here. If all of your terms are happily mapped to the perfect matches, then you won't need to do anything.

The second file is the Retired concepts file. This file contains all concepts that have been retired (by subset) from the Terminology Service along with any potential replacements for those retired concepts and the type of replacement (SAME AS, REPLACED BY, MAYBE A). There is a 30 day grace period between the time that the Terminology Service retires concepts and the time that the NAHLN server starts rejecting messages sent using the retired concepts. So you have 30 days from the time the NAHLN server

receives the Retired Concepts XML from the web service (and forwards it to you) to use this file to remove any mappings you may have to concepts that are retiring. Compare the list of retired concepts to your mappings, if you find any of them within your mapping, you'll have to re-map those concepts.

Remember when we talked about the two types of substitution? [See section on "corrections" in SNOMED.] This is where they will come into play. For the one-to-one "deterministic substitutions", there will be a single replacement concept for each retired concept, and it will have a "SAME AS" or "REPLACED BY" relationship with the retired concept. In these cases, all you have to do is replace the retired concept with the single replacement concept in your mapping. But there will, on occasion, be "ambiguous substitutions" where the relationship between the retired concept and the replacement concept is "MAY BE A" and more than one replacement concept might be given for each retired concept. With these, you will have to go back to the beginning and determine EXACTLY what your lab means when it uses that term. Hopefully one of the replacement concepts listed will match your usage and you can substitute the appropriate replacement concept, if not, it's time to visit the Terminology Discussions forum to post a request for help.

There will also be times when a concept is retired with no replacement concept given. These are most likely concepts that were being used to "organize" hierarchies, but were never really intended to be used anyway and have been removed because they were invalid. If you find that you have mapped to a concept that was retired with no replacements, post your need on the Terminology Discussions forum and the terminologists will help you find your way.

### Terminology Web Service

If you want to access the web service directly, rather than relying on the emails from the NAHLN server, you can. You may access a web form in the 'Updates' section of the Terminology Service website that will allow you to input a time frame (start date/time and end date/time) and optionally a given subset. The form will post to the web service and return the requested update in either XML or CSV format.

If you are REALLY adventurous you can also make your own call to the web service. The wsdl file for the web service is located at <https://vtsl.vetmed.vt.edu/nahln/ws/dataupdate.cfc?wsdl>. (Contact VTSL for further information). For now, the web service is open to all. If or when restrictions are applied to access, those restrictions will be posted on the Terminology Discussion forum, so stay in touch.

Issues have been discovered between .NET (and possibly other languages) and the Java based engine that drives the web services. It seems they disagree on how (or how not) to define XML document objects. Long story short, you can't call the web service above using .NET and retrieve an XML document. We have created an alternative web service that returns the XML as a string rather than a document object. It can be accessed at

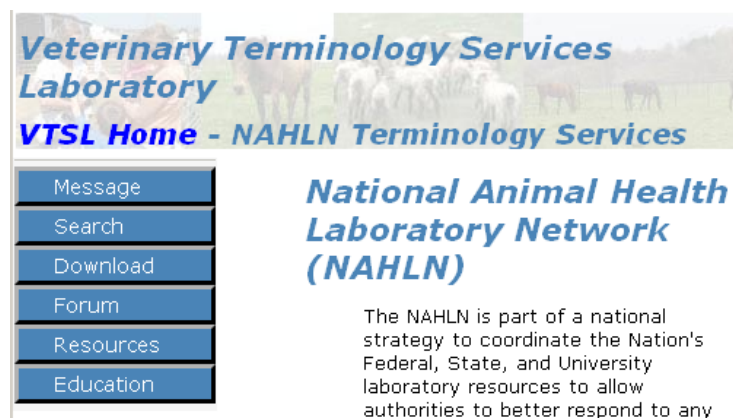
<https://vtsl.vetmed.vt.edu/nahln/ws/String/dataupdate.cfc?wsdl>. All parameters are the same, only the return type is different.

So what is this “Terminology Service” we’ve talked about so much in the preceding sections?

## ***Using the NAHLN Terminology Service***

The NAHLN Terminology Service is a project developed by the terminologists at Virginia Tech. It is a web-based tool which allows you to browse a visual representation of the message, as well as browse and search the terminology lists. It also provides access to the NAHLN discussion forum, a great place to go to ask questions of the terminologists as well as other labs who have already begun to message. This site is a work in progress, so please feel free to send suggestions to [jmgreen@vt.edu](mailto:jmgreen@vt.edu). The URL is:

<http://vtsl.vetmed.vt.edu/nahln/main.cfm>



The ‘Message’ button will take you to the visual representation of the message. This web version of the message can be toggled between a simple view that only includes the basics and a detail view that displays more details about each field (including things like repeatability, required or not, datatype, etc.). The simple view will suffice for most users. In this view, if a field has terminology or a standard list associated with it, there will be a hyperlink that will take you to the appropriate list.

The ‘Search’ button allows you to search the terminology lists. You first select the list in which you want to search for a concept and then type your search string. You can also click “View all” to display the entire list.

The ‘Download’ button is the place to go if you want to have a local copy of any of the lists or are looking to update the lists you have. Full subset files download as zip files containing lists in comma-separated value format. Updates can be downloaded in csv or xml formats.



The 'Resources' section is a links page that NAHLN users can share helpful links. You can view/use/upload links here that pertain to any aspect of implementing NAHLN messaging. Currently there are a few links there to helpful taxonomy sites for finding scientific names for birds and other mammals.

The 'Education' button takes you to a collection of presentations on SNOMED and its application. Some are useful background in their own right. Others address specific issues and might not make sense without knowing the historical context. But this is a good resource for those who want to know more. Note: These are the original presentations. Many links will be out of date but most materials referenced in the PPT and PDF files are still available if you go looking for them.

### Message Assembly

Once you have identified sources for all the data in the message, and have mapped those sources into standardized terminology, you are ready to assemble the XML message itself. In keeping with our philosophy of "Don't Panic," we won't go into the details of programming. But we just mention here that part of the reason the NAHLN uses the XML formatting of HL7 is the large number of tools available for constructing an XML "document" such as our message. Far be it from anyone in the NAHLN to try to tell you which of these tools is right for you. That would be the opposite of what we are trying to do by adopting standards. So how do you choose?

First, you want to survey the options. You can certainly code it out from the ground-up in your favorite programming language. For most of us, this wouldn't be the first choice, but it isn't that far-fetched. Especially with some scripting languages, this is right up their alley. Another option is to look for XML tools that work with your favorite programming language. Both Java and Microsoft's Dot Net have tools for what is called XML-binding. These tools take an XML schema and write software for you that will build and parse messages that conform to that schema. Programming using these generated bits of software is generally significantly easier and faster than starting from scratch. Or, you could skip programming altogether and simply format a report in the "shape" of the XML instance of the message as you wish to send it. Looking at a marked up sample message, you will see that the "fields" of actual live data that have to be filled in are relatively basic. This approach is less flexible than others but can be very quick and easy. The latest generations of database management systems (DBMS) have significant XML capabilities built in. It is possible to generate the message XML directly from the database using a special form of SELECT statement. "SELECT ... FROM ... FOR XML PATH(OPU\_R25), TYPE, ELEMENTS." One of the commercial LIMS that supports the NAHLN messaging directly uses this approach to efficiently generate some of their internal messaging.<sup>27</sup> Finally, you might chose to use a tool such as Rhapsody that provides a dedicated engine for mapping your data into the message structure. [See section on Rhapsody.]

---

<sup>27</sup> Thanks to Chris Westbrook at Computer Aid Incorporated (CAI) for this example using SQL Server 2005.

Once you know what tools you want to use, practice with them a little. Programmers call this writing the “Hello, World” program. But then don’t be afraid to jump in and try it. Generate a message with a sample of real or dummy data and see if it validates against the schema. Once it validates, try sending it to the test server, or even just posting it to the Implementation Issues forum on the Terminology Services web site. One thing you can be sure of is that no one before you did a perfect message the first time. The folks running the test server and on the IT Committee will be happy to help you work out the kinks.

### ***How Do You Know Your Message is “Correct?”***

In the *Hitchhiker's Guide to the Galaxy*, Slartibartfast tries to communicate to Arthur Dent that he must come with him or be killed.

**Slartibartfast:** You must come with me.

**Arthur Dent:** Who are you?

**Slartibartfast:** What? No. My name's not important. You must come with me, or you'll be late.

**Arthur Dent:** Late for what?

**Slartibartfast:** Well, um, what's your name Earthman?

**Arthur Dent:** Dent. Arthur Dent.

**Slartibartfast:** Well, late as in “the late” Dentarthurdent. It's a sort of threat. You see?

Here we have at least two common errors in communication. First, the word “late” gets translated from the intended concept to a completely different meaning of the same word. We also see a common problem with parsing and reformatting names. So, clearly, the NAHLN did not *invent* messaging errors. But it can be a frustrating effort trying to get that first message to be “correct.”

NAHLN message correctness involves a number of layers. In the “Tools of the Trade” section we already mentioned XML wellformedness and validity. Is the XML properly structured and does it contain all the elements required by the schema? This you can check locally using an XML editor that can do validation. But this only confirms correctness at the most basic scale.

The NAHLN base message schema includes many optional<sup>28</sup> elements that may be required by one or more testing programs. There are also “conditionality” rules in HL7 that are difficult or impossible to code in XML schema language. These kinds of things are checked on the server via “business rules.” A message that is valid according to the schema may still be sent back as structurally incomplete.

Finally, each of the coded value fields and some numeric fields have a set or range of allowable values for specific program testing result messages. These are also checked by the business rules. Using the terminology services for all coded values helps with this as

---

<sup>28</sup> Note for HL7 purists. We define these elements as Required or Empty rather than Optional. The fine distinction has to do with conformance with the standard. Required or Empty indicates that a compliant system must be able to handle these elements in a meaningful way if they are available in the source system or received from the sending system. Optional is really used to mean “we will deal with this in later refinements.” As program-specific schemas are developed from the base schema, some RE become R. The reverse never happens. So, to claim conformance at any level, the system must send all R elements and must be able to send any RE elements. We have no optional elements in the HL7 conformance sense, meaning that to claim full conformance, a product must be able to send all the elements.

does paying attention to specific program messaging requirements. Iowa State University has developed a message validation service that checks not only the structure of a message but also much of its data content. See <http://vdpam-vhis.vdl.iastate.edu/> for information about this service.

And, of course, the correctness of the message is only as good as the correctness of the underlying data in the LIMS. If the data in the LIMS are incorrect, incomplete, or delayed, so will the data in the message. Data quality is essential to modern laboratory operations and to NAHLN messaging.

Making a concerted effort to get the message right before ever sending it will do wonders at reducing frustration. Eventually, a round trip to the test server provides a final test of message correctness.<sup>29</sup> Once a sample message has been sent to the test server and returned an “AA” acknowledgement, you are almost there. [See Dealing with Acknowledgments.] It is still important to review the *meaning* of the data you successfully sent and ensure that it matches the actual test results. Finally, you can request a VS Program Review of your test message by sending an email as described in the *NAHLN IS User Account Management Guide* and program-specific messaging guides.

Of course, this final level of testing means you have to get the message to the test server in the first place, so we are a bit ahead of ourselves.

## Message Delivery

Congratulations! You’ve successfully constructed a NAHLN result message, populated it with all the right coded values and either saved it to a file or have it somewhere in memory. What now? How do you get this blob of text—or CLOB<sup>30</sup> if you are into correct acronyms—to the NAHLN repository?

## Protocols

The HL7 standard doesn’t say a thing about how to actually get the message where it is going. They leave that to the lower level “transport protocols.” This chapter will address these protocols and some of the tools that can be used to implement them. What, exactly, do we mean by a protocol? In intergalactic relations, most intelligent life forms like to maintain a degree of civility in their interactions. Vogons are a very notable exception, which is why Vogons are assigned as bureaucrats and not as ambassadors. To maintain this civility, ambassadors to other planets pay a lot of attention to the order of introductions, the greetings, handshakes—for species with hands—and other niceties of what is known as “diplomatic protocol.” When two computers meet in cyberspace, they rely upon similar agreed upon Internet port numbers, sequences of introductions, and handshakes to establish reliable and secure communications. These agreed sequences are

---

<sup>29</sup> To be honest, the Iowa State validation service will probably catch more than the official test site. Eventually a real person at APHIS will find anything you miss, so a trip or three to the Iowa State service is more than worth the trouble.

<sup>30</sup> Modern databases can store binary data in what are known as BLOBs (binary large objects). Some can also store large, unstructured—by the database—character data as CLOBs (character large objects). We are using “blob” in its non-acronym sense of an amorphous bunch of characters.

known as protocols. The Internet itself is not a place or thing, but the collection of these protocols.

The NAHLN messaging system uses a number of protocols to transport messages. This section will introduce you to these protocols. Once you have finished reading this you can forget the details and leave them to the tools we will introduce later in the chapter. Or for that matter, you may skip this section entirely and just take our word for it on the protocols.

### Identification and Authentication Protocols

“On the Internet nobody knows you’re a Vogan.” Or something like that. It is hard *not* to know how common it is for people and computers to impersonate other people or computers on the Internet. Who doesn’t get phishing email trying to get them to log into their bank and type their password? It has become an absolute rule that before sending or accepting any sort of sensitive data over the Internet you *must* verify that whatever is on the other end is whatever or whoever it claims to be. Computer security people divide this process up into two steps: “identification” is precisely defining what the other end claims to be; and “authentication” is having them prove it.

Before exchanging information such as that sent over the NAHLN messaging system, each end must identify and authenticate itself. We are all too familiar with user-id’s and passwords. The user-id’s are the identification part, and the passwords are the authentication. Passwords get a lot of bad press but, if used properly, they can form the basis of good authentication. Of course, part of using them properly is never divulging a password to an unknown system or person. So, if two entities on the Internet want to authenticate each other, they need something more complicated than just user-id’s and passwords or they would get in a vicious cycle, “Tell me your password. No, not until I prove you are who you say you are, give me your password first...”

The standard solution is for one end, usually a server computer of some sort, to use Public Key Cryptography to authenticate itself without having to divulge any secret. The details are very complicated to build, but very simple to use thanks to security standards. All we need is a copy of the server’s “certificate” and a trusted copy of the certificate that signed the server’s certificate. There are a number of different processes for establishing this initial trust (called “trust models”). All participants can agree to trust a single “root” certificate such as USDA’s Certificate Authority. Or each can explicitly trust the other’s signing certificate by exchange—via secure means—of a thumbprint. (See the section on “Security Tools” below for information on how this is actually done.) The system of trust that supports certificates is often described as a Public Key Infrastructure (PKI).<sup>31</sup> In any case, once each trusts the other’s signer, they can easily ensure that they are giving their password to the *real* NAHLN server, Terminology Service, or other system. (For even

---

<sup>31</sup> The year 2011 saw a flurry of security breaches caused by compromise of so-called “certificate authorities” also known as “CAs.” This was caused, at its root, by the explosion in the number of CAs that come pre-installed in our web browsers. Many of these are poorly regulated, housed in countries with dubious governments, etc. Manually assigning trust to a known good root such as the USDA Root Certificate Authority is a *much* more secure form of trust than accepting whatever comes in your browser. Sadly, this form of direct-trust is rare outside of the Department of Defense and other high-security environments.

more information about how PKI actually works to set up a secure connection, see “Transport Layer Security” below.)

Identification, authentication, access controls, data integrity mechanisms, and availability features combine to make the NAHLN messaging system “CIA secure.” This means that the right people can see the right information and the right people can add information, and *only* these people can do any of these things. How we define those permissions is a key factor in the design of the NAHLN.

### Role-based Access Control

Looking at each and every prospective user of the NAHLN messaging system and trying to figure out exactly what actions should be allowed and which should be prohibited would be an impossibly tedious task. And this tediousness would make it very error-prone. Instead the concept behind the NAHLN is to assign each prospective user to one or more “roles” that they play in the overall NAHLN system. So, one user may have the role of Area Veterinarian in Charge (AVIC), another as Federal Epidemiologist, another as State Veterinarian, and yet another as Diagnostic Laboratory Director. Even some non-human entities may have roles. An automated laboratory LIMS submission program that sends NAHLN messages may have the role of “File Upload User.”

Each of these *roles* is assigned a logical set of actions that it can perform. An AVIC may be able to view any results for samples submitted from his state no matter which laboratory did the testing, while a Laboratory Director can see any results of tests performed at her lab or referred on for additional testing from her lab. A VS program epidemiologist might be allowed to see all results anywhere in the country for diseases in that program while his regional counterparts would see only their own region. The automated submission process can only submit data.

Now, when a user logs into the system, they are allowed to perform those actions and view those data appropriate for any of their roles. In some cases, the positions of State Veterinarian and Laboratory Director are held by the same person. This person would normally be allowed to see all results for their state *and* all results of tests performed at their lab. There is a lot of overlap, but the two sets are not identical. In some systems—primarily medical and military—a user must take on only one of their roles at any given time. This is to ensure accountability for actions performed in one role vs. the other. The NAHLN uses the more normal combination of all assigned roles.

Role-based access, while simple in concept, is one of those areas where technology and policy come into close, unavoidable contact. It is essential for the policy makers clearly define the policy and for the technologists to reflect the true intentions of that policy. This can involve multiple trips by each into the other's comfort zone. But the result is a system that makes appropriate access seamless and inappropriate access impossible. It keeps things “CIA secure.”<sup>32</sup>

---

<sup>32</sup> True role based access has never been fully implemented on the VS NAHLN system(s). It remains a *conceptual* part of the NAHLN Information System design.

## Low Level Protocols

We won't spend much time on low level protocols even though we use them extensively. This is because things like Internet Protocol (IP) have become so ubiquitous that we can safely take them for granted.

Internet Protocol defines the addressing and routing system used by the Internet to get raw data to the intended computer. We assume that your computer has a connection to the Internet and is set up to use IP. Domain Naming protocols allow you to specify a human readable name for a computer—such as `www.nahln.org`—and have it automatically translated into the numerical address system used by IP. We also assume you are set up for this.

Besides the computer, IP addressing includes a sort-of sub-address known as a “port.” Each port provides a “service.” Some well-known ports and services include port 80 for web servers and 25 for outgoing email service. These ports sometimes become an issue for corporate or university network users, because network administrators have the ability to block traffic on specific ports. Computers called “firewalls” are often placed between the Internet and these corporate and university networks. They may allow traffic going out on some ports and coming in on some of the same or other ports. Because allowing external computers to connect to systems inside the network carries security concerns, blocking these connections is a legitimate thing for administrators to do. But it can be a problem if you have a legitimate need to allow these connections. So, you may have to convince your administrator that you have a legitimate need to go out or allow others in through some port or ports.

Transport Control Protocol is so esoteric that not even Zaphod Beeblebrox can get very excited about it. Trust us, it *is* important. But you will likely never think about it. You might however once in a while see the acronym “TCP/IP” in use. This protocol helps make sure that when a higher-level protocol sends a bunch of data, it all arrives and all gets put back together successfully. Now forget about TCP/IP and let the computers handle it.

## Hypertext Transport Protocol

Hypertext Transport Protocol (HTTP) is best known as the protocol by which web browsers request and receive web pages over the Internet. This involves one computer sending the other a request to “GET” a specific document. Most documents retrieved by GET calls are in Hypertext Markup Language (HTML) but there is nothing that says they have to be. In fact, part of the HTTP protocol is telling the receiving computer what kind of “file” is being sent. For the NAHLN, we use XML which you've seen earlier in this book. But wait! There's more!

The opposite of GET is “PUT.” PUT has mostly been replaced by POST for sending (posting) data to ordinary web pages, PUT is still useful for some specialized uses. PUT was originally designed to allow putting new pages on a website. That function has been largely replaced by more sophisticated file management systems. POST can be used to fill forms on the Internet. It can also send large chunks of HTML or XML. HTTP PUT,

POST and GET are used for applications to exchange data in what is known as Representational State Transfer (REST). The server receives the data, does what it will with it, and replies with a document just like that from a GET call. In the NAHLN messaging system, we need to PUT or POST a message to the repository and receive an acknowledgement back. When we send the acknowledgement in the response to the POST, we call this a “synchronous acknowledgement.” See a later section on the relative benefits of synchronous versus asynchronous acknowledgements. The HTTP POST protocol is just about exactly what we need to do. In this sense, the NAHLN repository will be acting like a web server, and our software will be acting like a web browser. But don't worry. We aren't going to ask you to read the acknowledgements in Internet Explorer. We'll get to tools for message transport later.

### Transport Layer Security

One of the design principles behind the Internet is the concept of “layers.” Each protocol uses the services of lower-level protocols. The result is a “layered look” of protocols on top of other protocols. Once this layering is set up, we the users only have to interact with the very top layer. One system for describing these layers is the Open System Interconnection model (OSI) which describes networking in seven layers. The top layer is what gives HL7 the “seven” in its name. Our HL7 seventh layer travels via HTTP in a lower layer. This eventually sits on a TCP/IP layer. And that travels over wires using even lower-level layers that we won't even discuss.

The problem with the stack of layers we've described so far is that anyone on the network can conceivably pick up the message and read its contents. Or worse, they might read the user identification and passwords used to connect to the NAHLN repository and send fake data. Or they could impersonate the NAHLN repository and receive all our data. One solution would be to start from scratch and string our own wires all over the country that only the NAHLN had access to. This is obviously not a workable solution. Instead, what if we could put a layer in there somewhere that would confirm the identity of the server and make the data unreadable to any but legitimate users?

Transport Layer Security (TLS) provides just such a service. This protocol was originally designed by Netscape and went by the name of Secure Sockets Layer (SSL) and the SSL acronym is still widely used even if slightly incorrect. This service provides a layer that looks to higher-level protocols exactly like a

### Sockets

The term “sockets,” as used in the old name “Secure Sockets Layer,” has an interesting history. Before the Internet was taken for granted, connecting via TCP/IP was a very complicated project for computer programmers. They had to write code to the technical TCP/IP specification for each program that needed to connect over the Internet.

On the UNIX platform where the Internet originally took form, a library of code was developed to help hide the messy details of TCP/IP programming. Programmers could “plug in” to this library. The library took the name “sockets” to reflect this.

Windows originally fought against the TCP/IP protocol in favor of its own proprietary networking. Outside suppliers developed libraries called “winsocks” or Windows Sockets that acted almost the same as UNIX sockets. Now, of course, Windows supports the Internet internally and still provides services known as winsocks.

normal TCP/IP sockets service. But, as discussed earlier, the implementation of this layer uses a technology called “public key cryptography” to ensure that the data are hidden from all but the intended system. Besides authenticating one or both ends, PKI provides encryption of the data as they flow over the wires.

As with all technology, and especially security, the TLS protocol has evolved over the years with more secure versions replacing less secure predecessors. The current version of TLS is labeled 1.3. The LMS server supports version 1.2 and up. This will inevitably change over the years and our networking stacks will need to be upgraded to comply.

Cryptography is the process of taking readable data and converting it to something that cannot be read at all except by someone with a piece of protected data called a key. The simplest form uses two identical keys. The first is used to encrypt the data. The same key or an exact copy is later used to decrypt the data back to its original form. This process is relatively fast and there are very good algorithms for keeping it secure. The problem is keeping the matched keys secure. How do you get them out to everyone who needs one without the bad guys getting copies as well? One solution is to use two different keys. One key must be kept very secure. It is “private.” The other can be sent to anyone. It is “public.” The public key can encrypt data such that only the holder of the private key can read it. It would take a rather thick book to go into details. Bruce Schneier's *Applied Cryptography* is a very good choice if you are really into that stuff. For now, let's just trust that they can use this one-way cryptography to establish reliable communication and to positively identify the holder of the private key. This type of protocol is known as public key cryptography.

To ensure the identity of the holder of the private key, you need a copy of the public key. That part is easy, it is public. But you have to be *sure* that it is the right public key. That part is harder. This is accomplished by a public key certificate. These certificates use similar cryptography to digitally sign off on the authenticity of the public keys contained inside them. You then have to trust the signer of the certificate. And we are once again into the snake eating its own tail business. Ultimately, you end up directly trusting a certificate authority (CA) such as Verisign, Entrust or Thawte that is in the business of signing certificates, or an organizational root CA such as USDA's. This is called “public key infrastructure” and can get very complicated. You can also skip that part and get your certificate directly from the system you want to trust. They simply sign the certificate themselves. In this case, you have to use a system you inherently trust to get your hands on the certificate. But once you've done that, you trust it directly. This is called a “direct trust” model. It is as secure, or more so, than a public key infrastructure but is harder to do on a very large scale. The NAHLN messaging system now uses a public root CA but earlier iterations used either a direct trust model or an organizational root CA.

Once again, we told you that so that we could tell you that you can almost forget about it. We will hear a little more about certificates later, but for the most part you can simply take comfort knowing that your data are being protected by transport layer security encryption.



## Hypertext Transport Protocol Secure

Because we are using a higher-level protocol (HTTP) over this lower layer security mechanism, we don't deal with TLS directly. Instead we use a variation of the HTTP protocol that shows that it is using the lower-level TLS. We call this Hypertext Transport Protocol Secure (HTTPS). Look at the address bar of your browser then next time you are buying somebody's used towel on eBay. You will see that the "http://" has changed into "https://." That is how simple it is to use all this cryptography stuff.

So, the simple conclusion of all this complicated background is that the NAHLN messaging system uses the HTTPS protocol to transport its messages and return synchronous acknowledgements. You can either learn all the details of each of these protocols, or choose a tool that has these protocols built in. For sake of this discussion, we will assume you will choose the tools approach. And that is the topic of the next section.

## Accounts and Passwords

**Trillian:** Who are you?

**Arthur:** Er, Dent, Arthur Dent.

**Trillian:** No, I mean "who" are you?

**Arthur:** Oh, the costume. Er, Livingston I presume. Yeah. Not as good as Darwin I know but the best I could manage at short notice.

**Trillian:** You're the first person whose gotten that right. Everyone keeps calling me Santa.

**Arthur:** Really?

**Trillian:** Yeah, and I thought the beagle made it a dead giveaway.

In real life—and in the movies, such as the movie version of *The Hitchhiker's Guide to the Galaxy*—who we are and who we are pretending to be comes across from things like out faces, voices, and costumes (and beagles). On the Internet we need more specific means of identifying ourselves and then proving that identity. Those of us who manage NAHLN messaging probably have at least three "accounts" to represent our specific identities. We have one account with the NAHLN to act on our own behalf. This identifies each of us as a person sitting at a computer interacting with the NAHLN. We also have a USDA eAuthentication account. This is a kind of super account that links our individual system accounts under a single identity. For systems that have been set up to use it, this allows us to access many different USDA systems with a single user id and password. Finally, we administer an account for the LIMS system to perform messaging on its own. We sometimes call this "system account" the "robot" account, because it tends to get as depressed as Marvin. That system has its own user id and password (or more recently EMS token).

## Requesting Accounts

To request system accounts you will need to submit an APHIS Form 513 through NAHLN.IT@usda.gov. You will likely need to repeat this process for each of the systems.

You will also need to obtain a level 2 eAuthentication account. This is a bit more complicated. You first go to the eAuthentication web site and complete an application.

<https://www.eauth.usda.gov/eauth/b/usda/registration>. Once you get your account information you may need to present proof of your identity to a federal office that can act as a local registration authority. This is usually a county FSA office or the state's VS Area Office. Proof of identity is usually a driver's license since those folks at DMV are the galaxy's most reliable judges of this outside of the Galactic Hyperspace Planning Council.

Things get a little more complicated with the system account. You apply for it the same way as individual accounts, but because your LIMS doesn't drive, it can't get an eAuthentication account of its own. Instead, it just gets a user id and password or token.

### Updating Passwords

To remain secure, passwords need to be changed from time to time. The logic behind password change frequency is complicated but it suffices to say that the more frequent the changes, the less risk of an imposter continuing to get away with pretending to be you. One very nice thing about eAuthentication is that, while you still have to change passwords occasionally, you only have to change the one eAuthentication password, not the individual account passwords.

The system account password or token has to be changed manually by contacting NAHLN.IT@usda.gov.

### Getting Help

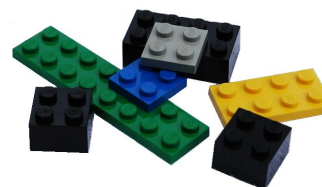
Sooner or later you *will* get locked out of one or more accounts. Help is available, but you have to ask the right people. The good folks at ATAC also known as APHIS Remedy (1-877-944-8457, [atac@aphis.usda.gov](mailto:atac@aphis.usda.gov)) may not know even what the NAHLN *is* but they have great ability to help with accounts and passwords. You just need to be very precise about the account and the system. When contacting ATAC by email or phone, keep in mind that they support all of APHIS. You will need to make clear that your question relates to Veterinary Services IT systems and specifically the NAHLN. For passwords, etc., they may be able to solve your problem directly, otherwise *expect* them to redirect your question to someone at VS. When contacting ATAC it is always a good idea to include the phrase "NAHLN-IT" in the subject line.

### Message Transmission Tools

The preceding discussion makes sending a NAHLN message sound pretty darned complicated. But by the same token, opening eBay and watching the little lock on the bottom of your screen close should be complicated too. The reason it isn't is that the tools do the work. In the eBay example, the web browser is the tool. You *could* use a web browser to send NAHLN messages, but that would mean doing everything one at a time. We'd rather have tools that let us automate the process. This section will discuss two of those tools. There are certainly others that implement the same protocols as described above. For example, the Java programming language includes libraries that make it almost simple to use these protocols and at least one lab has used Java to directly

send messages.<sup>33</sup> For now though we will look at two tools that are currently being used to transmit NAHLN messages.

These two tools represent two design concepts. The first is modeled after the UNIX operating system's design. The idea behind UNIX is that the system consists of a large number of small tools each of which does one job well and that integrates easily with other tools. Think of digital Lego<sup>®</sup> Blocks. A UNIX user or programmer can then build complicated systems through a simple process of connecting these small components. The cURL program is an example of one of these UNIX-like widgets that does just the transport part of our messaging system.



An alternate design strategy is to build a larger, integrated application that provides the user with “one stop shopping.” These applications which are popular in the Windows and Mac systems usually provide an integrated graphical user interface for the user, and often even for the programmer. The Rhapsody<sup>®</sup> integration engine from Orion is an example of this kind of system. In addition to the ability to implement the transport protocols used by the NAHLN, Rhapsody can do other communication protocols, track messages in and out, store log files, map local data to and from the message structure, and brew up something almost but not completely unlike tea. (OK, I made up the last part, but you get the idea.)

Neither of these approaches is necessarily better or worse than the other. They are just different. Which one you select depends on how you approach development, your budget, number of systems, expertise of your staff, etc.

### cURL

The human readable addresses you type into a web browser, etc., are known as Uniform Resource Locators (URLs). Traditionally, if you went to get a file, you'd use the File Transport Protocol (FTP) and an FTP client of some sort. But when you wanted a web page using HTTP, you'd use a web browser. Sometimes, you just want a simple utility to get the resource at the URL. This need gave rise to the “Client for URLs” which goes by the oddly capitalized acronym cURL. This is a little, command line tool for dealing with various types of URL connections. Command line tools seem un-user-friendly to many. But they have the great advantage of being easy to integrate into scripts, batch jobs, etc., in the spirit of UNIX. cURL has proven so useful in the UNIX environment that versions are now available for Windows, Mac, and Linux as well as all flavors of UNIX.

The command line for the NAHLN results for one lab<sup>34</sup> might look like this:<sup>35</sup>

---

<sup>33</sup> Arnon Jaya at Iowa State developed one such application. Michael Martin at Clemson developed a similar approach to the newer REST/POST method. This code can be obtained from the Terminology Forum on an “as-is” basis.

<sup>34</sup> Thanks to Jay Kammerzell, Colorado State University Diagnostic Laboratories for the example this was based on and description of their process.

<sup>35</sup> You may need to experiment with single (') vs. double (") quote marks around the userid and password fields depending on the kinds of special characters in your password. Single quotes seem to be the best in most cases in both Windows and UNIX systems. You also might need to double up the backslashes depending on whether your system sees those as escape characters.

## The Hitchhiker's Guide to NAHLN Messaging

```
curl -T 'C:\Outgoing\245DFFJ013006A.xml'
-H 'username: MyUserID' -H 'password: MyPassword' -H 'Expect:'
-D headers.txt -o C:\Incoming\245DFFJ013006Ar.txt
https://vsapps.aphis.usda.gov/HL7Result
```

This (as one long line) tells cURL to go to the URL <https://vsapps.aphis.usda.gov/HL7Result> and upload (pu-T) the file C:\Outgoing\245DFFJ013006A.xml. Since the web server is going to require a username and password, it supplies them as header fields (-H). It is then told to save the HTTP headers to a file headers.txt and the returned “page” of information to the file C:\Incoming\245DFFJ013006Ar.txt. Note that this script gives the results file a name related to the original upload file with just the added “r” and the type txt changed. This makes it easy to find which results go with which message.

For the test server the command line is slightly different. The “file” we are sending to is called “HL7ResultTest” instead of “HL7Result.” Otherwise, the command is the same:

```
curl -T 'C:\Outgoing\245DFFJ013006A.xml'
-H 'username: MyUserID' -H 'password: MyPassword' -H 'Expect:'
-D headers.txt -o C:\Incoming\245DFFJ013006Ar.txt
https://vsapps.aphis.usda.gov/HL7ResultTest
```

The only one who wouldn't go crazy trying to type a command line like that is Marvin the Manic Depressed Robot. But that is the whole point of this stuff. We are *trying* to make it possible for a robot (or a computer) to do it for us. Doing the above in a user-friendly graphical user interface would take a minute or more of clicking options and selecting files. Typing this and getting it right might take just as long, but the above example was *not* in fact, typed by a person. It was part of the output of a report generated from the lab information system when the results were ready. The report was in the form of a batch file (DOS script) that could be run when ready to send. The computer can do as many of these as there are results messages to send without making typos or getting bored.

A comparable command line for the new Enterprise Messaging System with its token authentication would look like this:

```
curl -d @message.xml -H @headers.txt -D ./headersOut.txt -o Output.txt
https://mrpvsmgs.aphis.usda.gov/PublisherService/PublisherServiceApi/ap
i/publish/lms -X POST
```

Here the headers are in a file to make it easier to manage the token string. headers.txt looks like this:

```
x-auth-token: D3DD80D . . . 455463E7768 (your token here)
Content-Type: application/xml
```

Besides installing cURL on a computer, all you need to make this work is the ability to write out your results messages to disk files and run a command line program or script.

Many of the other tools we use can perform many of these steps automatically. The workflow options are only as limited as your imagination.

### Rhapsody

cURL is a simple, single function tool. Rhapsody, on the other hand, is like a Swiss Army Knife.<sup>36</sup> You won't likely use Rhapsody to do just one thing, like handle message transport. You *could*, it just wouldn't make much sense. The point of a tool like Rhapsody is to help you integrate the various tasks that make up reading the data, assembling the message, finding the right recipient, sending the message, logging the message sent, receiving and logging the response, and notifying appropriate channels about success and error messages.



Rhapsody is a product of Orion Health Systems and product licenses are free to NAHLN laboratories and are available through USDA APHIS VS. The free license will allow the laboratory to create up to 10 communication points (Comm Points).

Rhapsody could be described as a process controller. If a certain condition exists as set up by the administrator then it forces the next action. Rhapsody starts and ends with a Comm Point. This could be a directory or a database query or an email. Comm Points cover a very wide range of actions.

Connecting the two points is a Route. A very simple Route with 2 Comm points could be this: Directory A → Directory B. If some process puts a file in Directory A, Rhapsody will monitor that directory according to intervals you set up with the route and once it finds a file meeting its criteria, it will automatically move it to Directory B.

Routes can be unidirectional or bidirectional. A bidirectional example of the Directory A → Directory B route might say take the file from A, process it in B, and return another file back to A to show the work was completed.

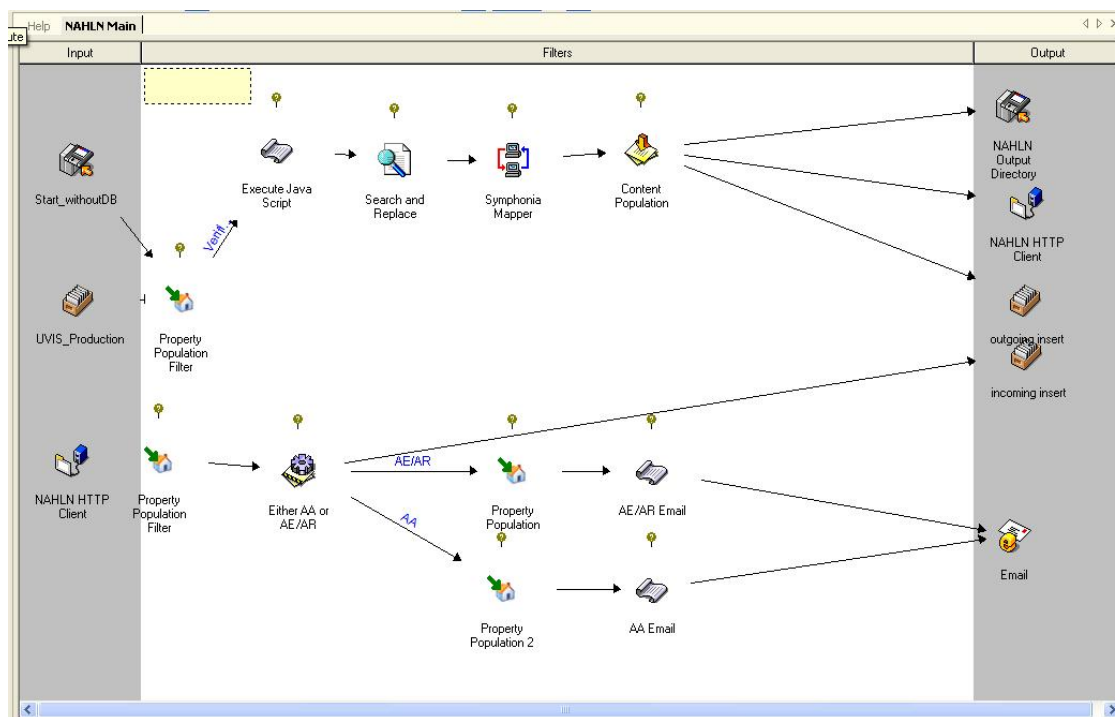
For use in the NAHLN, one Comm Point would be a query of your database to see if a NAHLN test was ready to be reported. The process would involve several routes and Comm Points. Once that condition exists, the query would be executed and the route toward NAHLN reporting started. From the query, we would need to map the data from your database into the xml file results message. Rhapsody has a tool called the "Symphonia Mapper" that assists in this process but it can still be very time consuming. You can probably get help with this by consulting with a lab that has mapped from your same brand of LIMS using Symphonia.

In another example, the Comm Point 1 query is executed and follows route 1 through the mapper and goes to Comm Point 2 which is the XML file output. That Comm Point 2 is the starting point of a new route which is the file transfer to NAHLN. The NAHLN

---

<sup>36</sup> Trademark and Image used by permission, Victorinox Swiss Army, Inc., 7 Victoria Drive, Monroe, CT 06468-1212

Comm Point will include logging into the NAHLN repository and getting the response message.



Another tool in Rhapsody is one which will log all the messages to allow you to see the successful messages and monitor the errors. From this tool, you could modify and resend a failed message.

Rhapsody allows for many types of error checking in between Comm Points and more advanced users can create very complex routes that will have a variety of ending Comm Points that could cover almost any situation.

### Security Tools

**Now that the NAHLN/LMS service uses a public root certificate you are unlikely to need any of the tools in this section. It may be removed or greatly reduced in future editions of this guide. Feel free to skip ahead to *Dealing with Acknowledgments*.**

In the earlier section on security we talked about “trusting a certificate” or “trusting a root.” These phrases represent trust in both the usual sense of trusting someone and in a more technical sense of telling the computer that it is OK to use that certificate or root to authenticate a connection.

There are two important concepts to be implemented in setting up a secure Public Key Infrastructure (PKI).<sup>37</sup> The easiest to understand is actually the complicated cryptography part. Any certificate is “signed” by some authority using a private key.

<sup>37</sup> It is much more complicated than this in real life, but most of the issues have been dealt with by other people such as those who have developed the tools we will be using.

Software can validate this signature to prove the identity of the authority and that the certificate has not changed since it was signed. So, all we have to do to trust a certificate is to trust the authority that signed it. This authority may prove its identity using a certificate; in which case all we have to do to trust it is to trust the authority that signed *its* certificate, and so on forever. Which leads to our second issue. At some point we must directly trust someone. Most systems come with some trusted authorities already installed. These are companies like Verisign and Entrust that have corporate reputations and so on. Since November of 2013, the NAHLN has used a commercial certificate issued by DigiCert. DigiCert, Verisign and Entrust do some validation of the entities to whom they issues certificates so you have a reasonable assurance that the `vsapps.aphis.usda.gov` really is what it says it is and not “hackers are us.” It is still a good idea to check its validity at some point.

The steps that follow are things that are most often skipped or glossed over by typical computer users. It is common to receive instructions that say something like, “If you get a security warning, just click OK...” However the sensitive nature of some of the data handled in the NAHLN means that it is well worth the few extra minutes to really verify that we only connect to the real NAHLN servers, Terminology Web Services, etc. The kinds of malicious attacks that these steps prevent are relatively rare, but do happen.<sup>38</sup> There are various ways that criminals can manipulate the Internet name system to get your browser or other software to go to the wrong place. Validating certificates will prevent this.<sup>39</sup>

We certainly don't want to be down in the weeds doing our own cryptography, and we don't have to. There are certificate tools present by default on most systems these days.

### Windows Certificate Manager

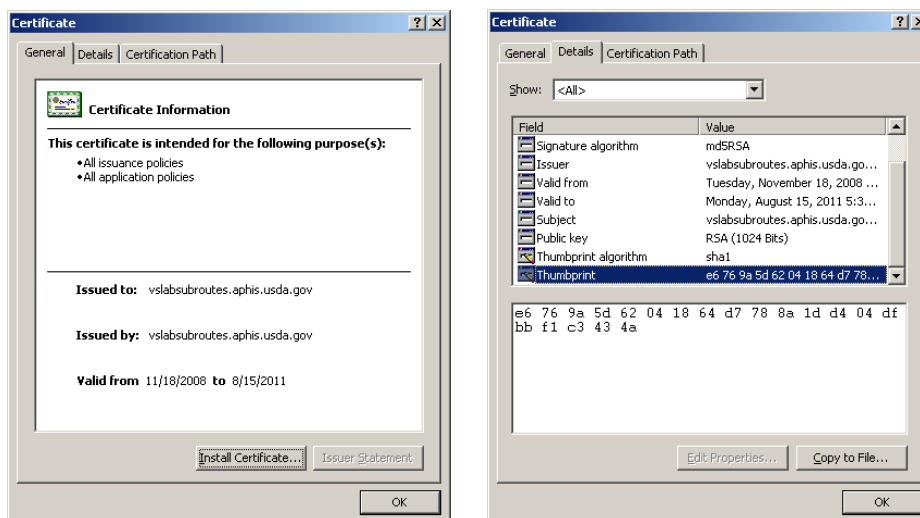
Windows comes with a certificate manager program as the default application for opening and working with files with extensions like “cer”, “crt”, and so on.

---

<sup>38</sup> As noted earlier, 2011 was a particularly bad year for PKI trust based computer security incidents. Getting the trust right *is* a big deal when it comes to real “CIA” security.

<sup>39</sup> Yet another way the bad guys fool computer systems is to gain access to one of the PKI root certificates that come “built-in” to Windows or your web browser. Having these automatically trusted is a great convenience but a significant security risk. It is actually *more* secure to manually assign trust as discussed in this section. High security systems remove all the automatically trusted roots as a security measure. We don't suggest going that far.

## The Hitchhiker's Guide to NAHLN Messaging



The two important features for a self-signed certificate are, the Thumbprint on the Details tab, and the Install Certificate button on the General tab. To be sure you have the real certificate and not an imposter, you want to communicate with someone at the issuing organization and have them give you the so called “thumbprint” for their certificate. This is a meaningless bunch of letters and numbers that have one and only one important feature. It is almost impossible to make a fake certificate that has the same thumbprint as the real one. So if you read this string of characters in Certificate Manager and it matches what your trusted counterpart gave you, you can be sure you got the right certificate. This is what we meant earlier by, “Software can validate this signature...” You can’t, of course get the thumbprint from the same web page, email, etc., where you got the certificate or *both* might be fakes. Security types call this process “out of band” because you haven’t gotten both through the same communication (“band” on a radio for example).

Assuming the thumbprints match—and they will, the likelihood of a forged NAHLN certificate showing up is *very, very* small in real life—you are ready to make the certificate part of your computer’s stored list of trusted certificates. To do so you click the Install Certificate button on the General tab. This starts a “wizard” to help you install the certificate. For the commercial NAHLN certificate, you can let it choose the folder.

The vsapps certificate can be viewed from the root of the server's URL, <https://vsapps.aphis.usda.gov/HL7ResultTest>. You will get just an error Acknowledgment Message as XML displayed in the browser but that will let you use your browser to grab the certificate.

### Mozilla Firefox

You can view a site's certificate in Firefox by clicking on the little lock icon and selecting “More Information” and “View Certificate”.

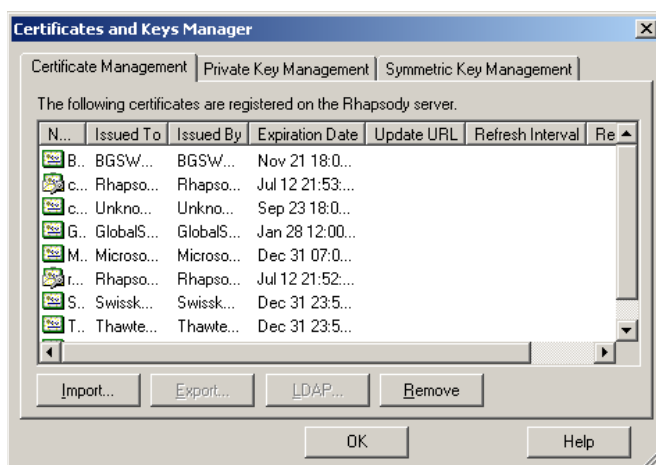


## Internet Explorer

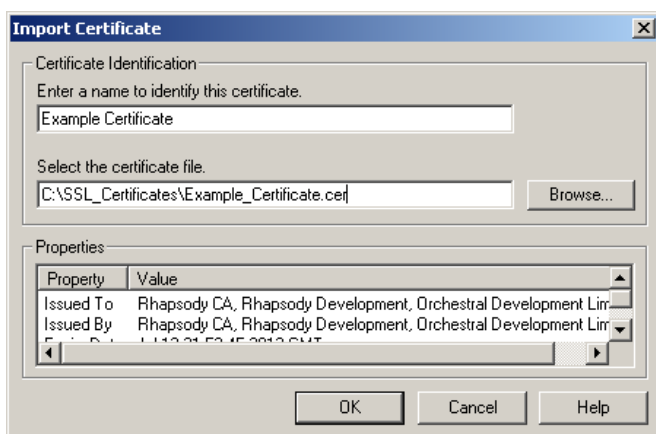
You can view a site's certificate in Internet Explorer by right clicking in the body of the page and selecting "Properties" and then "Certificates."

## Configuring Rhapsody to use Trusted (SSL) Certificates

1. Open the Rhapsody Administration Console
2. Import the SSL Certificate to Rhapsody's "SSL Certificates Manager"\*:  
 \* You can skip to step 3 if you have already imported your SSL certificates into the Rhapsody "SSL Certificates Manager"
  - a. Click View on the Menu toolbar then select "**SSL Certificates Manager**" from the drop-down list.
  - b. In the Certificates and Key Manager window, click "**Import**"



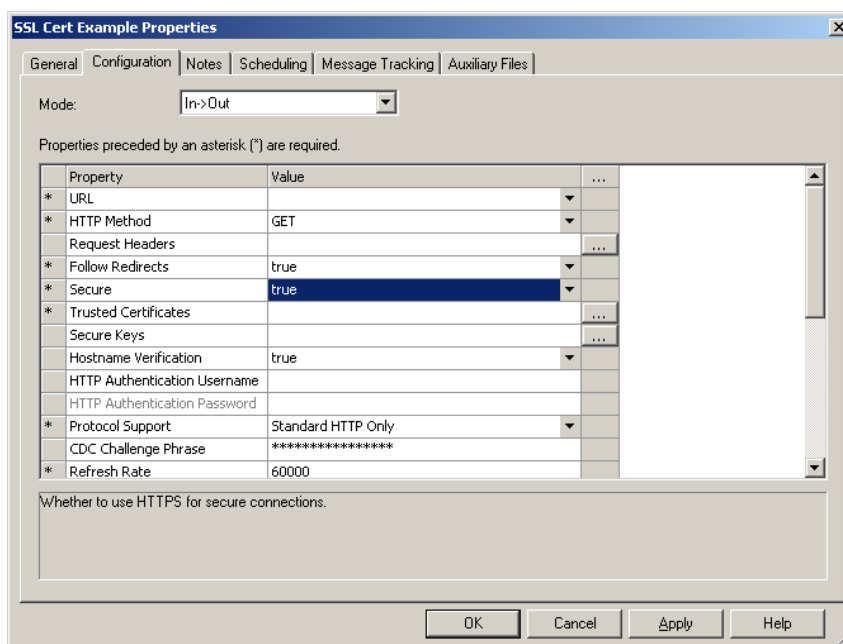
- c. In the Import Certificate window, enter a name for the certificate identity and then click browse to select the certificate




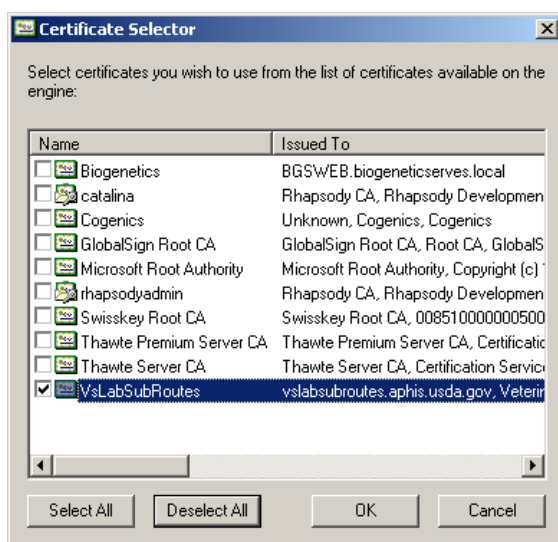
- d. The certificate added above should now show up in the list under the Certificate Management tab.
  - e. Click OK to close the SSL Certificates Manager.

## The Hitchhiker's Guide to NAHLN Messaging

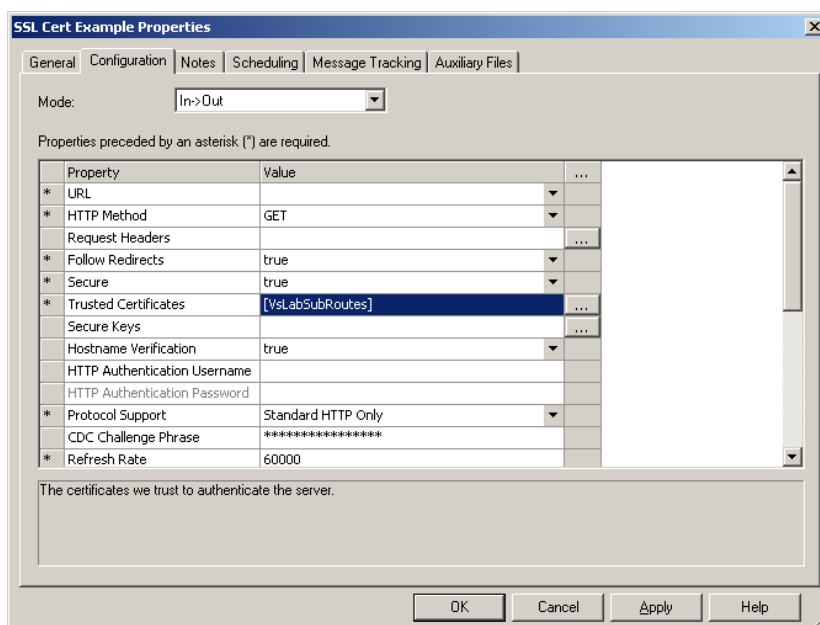
3. Now find the HTTP Comm. Point in Rhapsody (or create one if you haven't already) you would like to add a trusted certificate to and double click on it to bring up the HTTP Comm. Point properties window.
4. Click on the Configuration Tab, go to the "**Secure**" property and change it from *False* to *True*. The "**Trusted Certificates**" property directly below should now go from being grayed out to editable.



5. Click the  area to the right of the "**Trusted Certificates**" property which will open the "Certificate Selector" window. Now select the appropriate SSL certificate from the list (if you added a new SSL cert in the steps above, select it from the list) and click OK.



- The certificate you selected above will now show up in the “**Trusted Certificates**” property. Click Apply then OK on the Comm. Point properties window to save the changes.



- Check in the Comm. Point and you've completed the steps to add a Trusted SSL Certificate into your Rhapsody route.

## Configuring cURL to Use vsapps Certificate

First, obtain the file curl-ca-bundle.txt from someone you trust at the NAHLN. Rename this file curl-ca-bundle.crt and place it in the same folder as your cURL executable file. cURL will automatically look in this folder for trusted certificate bundles.

## Java Keytool

For Java programming the trusted certificates are handled by a small application called Keytool. Its use is beyond the scope of this guide but can be found at <http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/keytool.html>

## Dealing with Acknowledgements

The subject of Acknowledgements in HL7 can get very complex. Fortunately, the NAHLN uses a very simple form of acknowledgement. Use of the NAHLN Messaging protocols with other labs or states may involve a slightly more complicated acknowledgement sequence as discussed later.

The HTTP protocol that the NAHLN uses to send result messages results in an HTTP Response. If you were browsing the web, this response would be the web page. When you send a NAHLN message, the response is the Acknowledgement message known in HL7-speak as an ACK. This is a simple bit of HL7 encoded as XML. The HTTP Response also includes a header. When browsing the web, you don't normally see the

header unless you type in a web address wrong; then you see “404 page not found” or something like that. The mechanism you use to send NAHLN results must have some way of dealing with all these responses.

We can group possible responses into four types; one at the transport level and three at the message level. If the header comes back as anything other than “100 Continue” followed by “200 OK” then something has gone wrong with the HTTP connection itself. In this case, the message was never even sent. The response is to fix whatever is wrong with the connection—which might just mean waiting and trying again later—and resending the message. This is the *only* situation in which a message can be resent with the same message control ID. Actually, since it wasn't sent in the first place, this is just sending it the first time. This may be used with tools that, for example, retry after an HTTP time-out. If any significant time takes place between tries, the message should be regenerated with the correct message date/time. In that case the message control ID can be reused, but can also be replaced with a new system generated ID.

If we get a response body—the ACK message—we must read the XML to determine the status of our message. For now, there are only three fields in the entire ACK that matter to us. The first two are in the MSA segment. MSA.2 gives the Message Control ID of our message. This can be useful in pairing up ACKs with their original messages after the fact. MSA.1 is the acknowledgement code that tells us whether the NAHLN “liked” our message. We have already discussed reasons messages might not pass the NAHLN's error checking. If we get an “AA” which stands for “Application Accept” in MSA.1, we are good. If we get an “AE” which stands for “Application Error,” we have a problem with our message. In this case, a third field becomes important. The field ERR.7 gives a human readable description of the error. These are generated by the server, so “human readable” is a relative term, but it is usually enough to help figure out what went wrong. Even if this error description is unreadable to you, it is worth saving in case you need to request help on the forum. The contents may be more useful to one or more of the people who monitor that resource and provide help. If you get an “AE” acknowledgement, you need to fix the problem and send a new message with the same result or results as the message that was in error.

A special case has been added to the ACK to support the system account password change requirement. As the system account password nears its expiration date, you will begin to get ACK with MSA.1 containing “AA” but with an ERR segment. The severity (ERR.4) will be “I” for “information” and the message in ERR.7 will announce the expiration date of the password. If the system password is allowed to expire, it will need to be reset by ATAC, so watching for this “warning” will be important.<sup>40</sup>

There is a third acknowledgement code that we need to address. If the response comes back “AR” which, in an odd choice of words, stands for “Application Reject,” we have some digging to do. In general, this response means that something was wrong on the server end such that the message could not be completely processed. But it *was* received.

---

<sup>40</sup> This is another of those features that has not yet been implemented on the NAHLN server. But should be eventually. Other types of information and/or warnings may also be coming in the future.

In this case, we usually just wait a while and resend. But we cannot send the same message. We need to generate a new Message Control ID and Date/Time of Message. The difference between this and the reaction to the “AE” response is that we don’t need to change any of the message content, only the contents of the MSH segment. The easiest way to do this is usually just to change the status of the result or results in the message back to “unsent” and let the same message contents be regenerated in the next cycle. If you repeatedly receive “AR” results, you should report the error to NAHLN-IT@usda.gov.

The whole point of this processing is simply this: Each message as defined by the message control ID should reach the NAHLN system only once. Each result as defined by your system should reach the NAHLN and return an “AA” acceptance acknowledgement once and only once.<sup>41</sup> For convenience, especially in accessions with multiple tests, the same result may be resent as long as the value and observation instance identifier are identical. For each “AR” or “AE” response, you need to generate a new message with a new message control ID, and errors corrected as needed. This process continues until you get an “AA.”

### ***Deprecation Warnings***

Keeping senders and receivers of messages working seamlessly together can be tricky. When changes are needed to the message structure or terminology codes are retired or replaced, there must be a grace period in which the receiver prefers the new structure and codes but will accept the old ones until the sender can update their structure or codes. Such content is called “deprecated.” When a message uses either structure or codes that are still valid but considered deprecated, the receiver may return an acknowledgment code of “AA” but still include an ERR segment with a warning. The severity code in ERR.4 would be a “W” for “Warning.” The error code in ERR.3 would still be “0” for “message accepted” but the application error code in ERR.5 would be “D” for deprecated usage.

### ***Acknowledgements When Using a Message Router***

Some message routing services can provide a service where they store the message temporarily and allow the receiving end to “call in” to collect any recently received messages. Obviously, such a service cannot return an application acknowledgement to your sending system in the original connection. In this case, there is a system of “commit acknowledgements.” When sending via a router to a system that uses this “polling” protocol, you will receive one of three acknowledgment codes: “CA” for “Commit Accept”, “CE” for “Commit Error” or “CR” for “Commit Reject.” You respond to these exactly as you would to “AA”, “AE”, or “AR.”

The only real difference is in “CA.” This means that the router received the message and sees nothing wrong with it. The router takes responsibility for getting the message to the recipient. But there is a chance that the message will fail one or more fine grained rules

---

<sup>41</sup> This “feature” has caused problems for some systems based on the weird way VS handles the results once they are received. For some programs results may be entered via another system such as VSLS and be rejected by the NAHLN messaging system because they already exist. This will hopefully be resolved in the latest implementation.

that the router does not, or cannot check for you. In those rare cases, you may initially receive a “CA” response but later get an “AE” error from the ultimate destination. Receiving these “asynchronous” application acknowledgements requires your system to be a message receiver. This can be set up as either push or pull style.

### **Messaging Details for Specific Test Programs**

The general *Implementation Guide* and the discussion up to now in this *Hitchhiker's Guide* relates to messages relevant to any of the NAHLN's testing programs. A lab that can send the full message will be able to handle the details required for *any* of these testing programs. However, not all information is available to labs for every testing program. The information needed to manage these programs may have been entered elsewhere such that all the lab needs to add are the results themselves plus enough identifying information for the NAHLN repository or disease management program to reunite the results with their case information.

This chapter will detail the changes to the message structure for each testing program. These are not *technically* changes. Things that are required are still required, etc. A message that validates against the main schema must also include correct data in program required fields. Many of the “Required or Empty” elements will end up required by business rules or conversely the data for these fields may be explicitly excluded<sup>42</sup> so that the elements will always be empty for these elements or made non-repeatable.

The NAHLN now provides program-specific messaging guides that provide very detailed information about the program requirements. This chapter will not duplicate that information. Instead, we will dive into the logic and implementation of some of those details that may not be immediately obvious from the program guides. Other parts of the material here is for test types that are not currently in production use. It is hoped that they will demonstrate the degree of flexibility built into the basic NAHLN message structure that gives it the ability to handle virtually any veterinary diagnostic laboratory testing scenario. This flexibility allows programs to collect the data needed by any given situation. The current version of each program's messaging guide will always be the most current source for program requirements.

### **Wildlife Services Avian Influenza**

NAHLN messaging of Avian Influenza (AI) testing introduces a few variations on the generalized NAHLN Results Reporting message. Some of these are based on laboratory procedures and others on programmatic requirements. But even with these required variations, the basic message structure and guidelines remain. This section will detail the variations along with the rationale behind them.

### **Minimal Message Content**

---

<sup>42</sup> This is not technically a proper constraint. Cases in which specific information is disallowed will be minimized but may occur in situations where a program has agreed to *never* collect a given piece of information. In those cases the LIMS should not contain the information anyway, so the field would be empty even if you tried to send it.

Most of the information that would normally come into the lab in the form of an order message and be repeated in the result message instead goes directly to the Veterinary Services Laboratory System (VSLS) via a web-form. As long as our results can be paired up using the included identifiers, most of the demographics, submitter and collector, etc., can be omitted from the result message itself.

### **Pooling**

We introduced the idea of pooling earlier in the confusing concepts section. That discussion involved pooling from multiple animals in order to test one common specimen, usually to save time or money. Wildlife Services' AI testing uses a different sort of pooling. In this testing program, oropharyngeal swabs are pooled with cloacal swabs in the field. For purposes of the NAHLN these field-pooled specimens are treated as individual specimens with a unique specimen type designated by the SNOMED code 30931000009104, "Pooled cloacal/oropharyngeal swab (specimen)". Because each pool comes from a single bird, no attempt is made to determine which swab caused the positive result. Even though the specimen type is "pooled" the specimen role is "patient" because one animal is being tested.

### **Programmatic Issues**

As with many animal health regulatory programs the Avian Influenza surveillance program deals with much sensitive information. Some data that would ideally be included in the messaging is up to the submitter to hold and provide to the surveillance program by other routes.

### **Wildlife Services Submission Form Mapping**

One effective approach to generating a NAHLN result message for program testing is to start with the program-specific submission form. First map the fields on the form to the corresponding fields in the message. Then the task is just to track each of these fields as they flow through the LIMS. It may be obvious where each ends up. But sometimes, especially with an older, less standards-aware LIMS the lab may have had to resort to putting data in unusual places. If this work-around is needed, it is critical that the data-entry staff be well trained to be absolutely consistent in their usage of such data fields. Watching the data-entry process closely is one way to begin the tracing. Once the data are tracked to the LIMS database or export format fields, all that remains is the mapping to appropriate coding systems.

## The Hitchhiker's Guide to NAHLN Messaging

02759

USDA Wildlife Avian Influenza Surveillance Data Sheet APHIS Page 1 of 7

Collector: Sheldon Owen Testing Laboratory: Clemson Diagnostic Please charge to purchase order #: AG-6395-P-08-0288  
 Agency: USDA WS City: Columbia State: SC  
 Phone number: 803-210-9937 County: Clarendon State: SC

Referral #: SK 10 08 28 08 3 most abundant species on site: WOODU  
 State, initials, month, day, year  
 GPS location (In WGS 84 and decimal degrees): N 33.55987 W - 80.44616 Collection Site: Santee NWR

Date collected: 8/28/08 Collection Strategy (circle one): live bird Hunter Harvest Agency Harvest Sentinel Morbid/Mort Environmental

| Sample Bar Code  | Bird Species Code | Sex:                                      | Age Class  | Sample Type:   | Comments:                            |
|------------------|-------------------|---|--|--|--------------------------------------|
| <u>A00520761</u> | <u>WOODU</u>      | 1. Male<br>2. Female<br>3. <u>Unknown</u> | 1. Hatch Year<br>2. After Hatch Year<br>3. <u>Undetermined</u> | 1. Cloacal<br>2. Tracheal<br>3. <u>Oral + Cloacal</u><br>4. <u>Fecal</u> | (Subject ID, band #, condition, etc) |
| <u>A00520762</u> | <u>WOODU</u>      | 1. Male<br>2. Female<br>3. <u>Unknown</u> | 1. Hatch Year<br>2. After Hatch Year<br>3. <u>Undetermined</u> | 1. Cloacal<br>2. Tracheal<br>3. <u>Oral + Cloacal</u><br>4. <u>Fecal</u> | (Subject ID, band #, condition, etc) |
| <u>A00520763</u> | <u>WOODU</u>      | 1. Male<br>2. Female<br>3. <u>Unknown</u> | 1. Hatch Year<br>2. After Hatch Year<br>3. <u>Undetermined</u> | 1. Cloacal<br>2. Tracheal<br>3. <u>Oral + Cloacal</u><br>4. <u>Fecal</u> | (Subject ID, band #, condition, etc) |
| <u>A00520764</u> | <u>WOODU</u>      | 1. Male<br>2. Female<br>3. <u>Unknown</u> | 1. Hatch Year<br>2. After Hatch Year<br>3. <u>Undetermined</u> | 1. Cloacal<br>2. Tracheal<br>3. <u>Oral + Cloacal</u><br>4. <u>Fecal</u> | (Subject ID, band #, condition, etc) |
| <u>A00520765</u> | <u>WOODU</u>      | 1. Male<br>2. Female<br>3. <u>Unknown</u> | 1. Hatch Year<br>2. After Hatch Year<br>3. <u>Undetermined</u> | 1. Cloacal<br>2. Tracheal<br>3. <u>Oral + Cloacal</u><br>4. <u>Fecal</u> | (Subject ID, band #, condition, etc) |

Date Samples Shipped to Testing Lab: 09/02/08 # of samples in referral: 31  
☒ Check here if collector info is same as submitter  
 Name of Submitter: Sheldon Owen Phone # of submitter: 803-786-5418  
 Please send all results to Dr. Thomas DeLiberto by email ([WSlabresults@aphis.usda.gov](mailto:WSlabresults@aphis.usda.gov)) or by fax to (970) 266-6215.  
 Call NVSL immediately at (515) 663-7551 with all positive H5/H7 rRT-PCR results. Results may also be entered in directly to the lab submission website: <https://cowebapps.aphis.usda.gov/vslabsub/login.do>.

Comments: \_\_\_\_\_ FOR LAB USE ONLY Revision 1/11/2008

WS AI Submission Form With Mapping of Required Elements

### Data Not Provided

Note that in the mapping image above that there are a number of fields not mapped. Collector information is available for the ROL segment for collector. Several additional fields for PID are present, including sex, age and conceptually the barcode as a patient identifier as well as a specimen ID. It would not be an error to map these and send in their respective fields, but it is unnecessary for the Wildlife Services Avian Influenza program because these data are all being entered as part of the electronic order entry in the VSLS system.

### Referral Number

The Wildlife Services referral number has presented special problems. The program has very specific requirements for the content of this referral number but is less strict on its format. Because computers get really hung up about things like spaces and two vs. four-



digit years, it is not infrequent for the referral number in VSLs to have the same content, but different format from that on the paper submission form and to, therefore, fail to match when the results arrive at the NAHLN server. Because this information is needed for human analysis of the results, but not strictly required to pair results with their electronic order, the NAHLN has elected to treat this identifier as if it were just another human-readable attribute. No effort will be made to match records based on the value of this field. It remains important for human use however. If the formatting issues are ever completely resolved, it may once again be treated as a full-fledged identifier.

### ***Influenza A Virus in Swine (IAV-S) Surveillance***

“The *Encyclopedia Galactica*, in its chapter on Love states that it is far too complicated to define. The *Hitchhiker's Guide to the Galaxy* has this to say on the subject of love: 'Avoid, if at all possible.' Unfortunately, Arthur Dent has never read the *Hitchhiker's Guide to the Galaxy*.”

The *Hitchhiker's Guide to NAHLN Messaging* warns that the IAV-S program is almost too complicated to define. But it provides the advice, “Don't Panic.” Each of the details of the program is easily handled by the NAHLN Result Message if we just keep the basic principles straight and take them one at a time. So, don't be like Arthur, and you will get through messaging IAV-S results just fine.

The swine influenza surveillance program introduces a number of elements that have not been used in earlier NAHLN supported surveillance programs. The testing protocols are more complex with the results of one set of tests analyzed to determine which if any follow-up testing to perform. The concept of sequencing viral genes with the “answer” being the actual sequence of bases was new even to the experts at LOINC. The current implementation returns the GenBank Accession Number as the “answer,” which better fits common LOINC coding. And sequencing is not performed on the original specimen but on the isolate created during virus isolation. So, we must deal with parent specimens. The IAV-S program also includes the concept of either anonymous or traceable testing so we need the ability to exclude or mask identifying information while still tracking specimens through the system.

### **The Premises “Role”**

One vital bit of information is the location from which the specimen being tested was collected. This is the epidemiologists famous “where.” In normal testing this bit of information is the location and identification—ideally Animal Disease Traceability program PIN or LID—of the farm from which the patient came. In traceable IAV-S testing this is still the case. But even in anonymous testing, we need to know the state and type—farm, market, lab, etc.—of location.

The Premises ROL segment can take two different forms for anonymous collection. The first includes only the required and relevant elements.

## The Hitchhiker's Guide to NAHLN Messaging

```
<ROL>
<ROL.2>UC</ROL.2><!--Code for unchanged. No changes are requested for these ROL data-->
<ROL.3>
  <!--The Role being played by the person or organization represented by this segment-->
  <CWE.1>PREM</CWE.1>
  <CWE.2>Collection Premises</CWE.2>
  <CWE.3>HL70443</CWE.3>
</ROL.3>
<ROL.4>
  <XCN.2>
    <FN.1>NA</FN.1>
  </XCN.2>
</ROL.4>
<ROL.11>
  <XAD.4>MN</XCN.4>
</ROL.11>
<ROL.13>
  <PL.6>P</PL.6>
</ROL.13>
</ROL>
```

The second form includes those elements that are explicitly NOT shared.

```
<ROL>
<ROL.2>UC</ROL.2><!--Code for unchanged. No changes are requested for these ROL data-->
<ROL.3>
  <!--The Role being played by the person or organization represented by this segment-->
  <CWE.1>PREM</CWE.1>
  <CWE.2>Collection Premises</CWE.2>
  <CWE.3>HL70443</CWE.3>
</ROL.3>
<ROL.4>
  <XCN.2>
    <FN.1>NA</FN.1>
  </XCN.2>
</ROL.4>
<ROL.11>
  <XAD.1>
    <SAD.1>MASK</SAD.1>
  </XAD.1>
  <XAD.3>MASK</XAD.3>
  <XAD.4>MN</XCN.4>
  <XAD.5>MASK</XCN.5>
</ROL.11>
<ROL.13>
  <PL.6>P</PL.6>
</ROL.13>
</ROL>
```

This version makes explicit that we know the address, city and zip code, and we are withholding the information for confidentiality.

Why would you send the more verbose second version? One reason may be simply that it is easier to alter the contents than the structure of your ROL segment. The other is that you want to make clear that you *do* have information on the actual location but have concealed it. (Or you may have an old schema that required these elements, but that is not a very good reason.)

For traceable testing this segment is used to convey the actual farm, market, etc.—here a market—where the sampling took place. This includes the ADT program PIN in PL.10/EI.1.

```
<ROL>
  <ROL.2>UC</ROL.2><!--Code for unchanged. No changes are requested for these ROL data-->
  <ROL.3>
    <!--The Role being played by the person or organization represented by this segment-->
    <CWE.1>PREM</CWE.1>
    <CWE.2>Collection Premises</CWE.2>
    <CWE.3>HL70443</CWE.3>
  </ROL.3>
  <ROL.4>
    <XCN.2>
      <FN.1>NA</FN.1>
    </XCN.2>
  </ROL.4>
  <ROL.11>
    <XAD.1>
      <SAD.1>123 4th Street</SAD.1>
    </XAD.1>
    <XAD.3>Bemidji</XAD.3>
    <XAD.4>MN</XCN.4>
    <XAD.5>56601</XCN.5>
  </ROL.11>
  <ROL.13>
    <PL.6>M</PL.6>
    <PL.10>
      <EI.1>MN1234Q</EI.1>
      <EI.3>2.16.840.1.113883.3.5.6.1.1</EI.3>
      <EI.4>ISO</EI.4>
    </PL.10>
  </ROL.13>
</ROL>
```

## Age Estimates

In order to facilitate analysis based on approximate date of birth, the IAV-S program has specified a system for converting age group to specific dates for the “Date of Birth” (PID.7) field. It is understood that these are just estimates of the average DOB for pigs in each age range. But by converting to average DOB, the program can track changes over time, etc. The formulas for various age groups are:

- Collection date subtract 30 days (nursery < 1 month old)
- Collection date subtract 60 days (suckling 1-3 months old)
- Collection date subtract 150 days (grower/finisher 3-8 months old)
- Collection date subtract 240 days (sow/boar > 8 months old)

The calculated date is to be reported as the date of birth in the HL7 date format YYYYMMDD.

Age group can also be communicated as an observation on the patient using the SNOMED codes discussed earlier and under PED below.

## Specimen Barcodes

All submissions for IAV-S testing are expected to have globally unique barcoded specimen identifiers. The value from this barcode is the Placer Specimen Identifier with an OID of 2.16.840.1.113883.3.5.8.5 for the IAV-S program. This may or may not still apply. As always the Filler Specimen Identifier is required and must be unique.

## Parent Specimen

Earlier we devoted an entire section to the idea of specimens having “parents.” The IAV-S program involves one of these cases. In the case when the results of the initial PCR testing require follow-up with virus isolation and sequencing, we end up creating a new specimen—the isolate—from the original swab, etc. The original swab is said to be the parent of the isolate. The isolate is the specimen for the sequencing. In order to be sure we can figure out which specimen's virus was ultimately sequenced we need to keep track of the isolate's parentage even as we assign it an identifier of its own.

```
<SPM.2>
  <EIP.1>
    <!-- Specimen barcode from original specimen (not 100% corr here) -->
    <EI.1>A01270342</EI.1>
    <!-- Assigned by the IAV-S program -->
    <EI.3>2.16.840.1.113883.3.5.8.5</EI.3>
    <EI.4>ISO</EI.4>
  </EIP.1>
  <EIP.2>
    <!-- MN LIMS's internal unique specimen ID -->
    <EI.1>179583</EI.1>
    <EI.3>2.16.840.1.113883.3.5.1.26</EI.3>
    <EI.4>ISO</EI.4>
  </EIP.2>
</SPM.2>
<!-- This specimen was derived from 179472 by culturing it. -->
<SPM.3>
  <EIP.1>
    <!-- Specimen barcode. Here tells the specimen this one was g from -->
    <EI.1>A01270342</EI.1>
    <!-- Assigned by the IAV-S program -->
    <EI.3>2.16.840.1.113883.3.5.8.5</EI.3>
    <EI.4>ISO</EI.4>
  </EIP.1>
  <EIP.2>
    <EI.1>179472</EI.1>
    <EI.3>2.16.840.1.113883.3.5.1.26</EI.3>
    <EI.4>ISO</EI.4>
  </EIP.2>
</SPM.3>
<SPM.4>
  <!-- It is now a new type of specimen appropriate for sequencing -->
  <CWE.1>621000009105</CWE.1>
  <CWE.2>Viral isolate specimen (specimen)</CWE.2>
  <CWE.3>SCT</CWE.3>
</SPM.4>
```

We do one rather unorthodox thing here. Because the submitter—“placer” in HL7-speak—has no actual identifier for the isolate, and because we want to make it easy for the submitter to identify results to source, we incorrectly label the isolate with the original barcode. The barcode or other label supplied by the submitter does correctly identify the parent specimen in this case.

## Order Group for Entire Protocol

In previous testing programs we saw a one-to-one correspondence between the order code in OBR and the observation code in OBX. Sometimes though, the order may actually represent a complex protocol involving a number of different “tests” or observations. Most of these are not common to both veterinary and human medicine so they do not have standard LOINC codes. But we can assign protocol specific order codes in the extension space of LOINC. These codes begin with an “X.” You will not find them at <http://search.loinc.org> or in RELMA, but we will put them on the Terminology Services website. The extension LOINC code for the overall IAV-S protocol is X0001-8.

In these cases the message will have a SPECIMEN group with one ORDER group. In the ORDER group will be an OBR, and ORC, and a series of RESULT groups each with its own OBX segment. The OBXs will be identified by LOINC codes for the individual observations such as Matrix PCR, H Typing PCR, etc. In general, it is acceptable to group like this all observations that return results at about the same time. Keeping with the NAHLN approach of reporting results as soon as they are finalized and verified, if a significant amount of time will pass before further results become available, it is best to message what you have and put the rest in a later message.

If results apply to two different specimens then the ORDER must be repeated because the ORDER is nested inside the SPECIMEN group.

## Reason for Submission

Previously in NAHLN testing very little was made of the “Reason for Submission” (OBR.31) field. We simply used “N” for “National Surveillance.” The IAV-S program is different. This program needs more detailed information on which surveillance stream produced each specimen.

| Code | Text  | Code System |
|------|---|-------------|
| G    | Diagnostic submission with clinical signs ( <i>for IAV-S- applies to diagnostic swine submission stream AND sick pigs at concentration points or commingling events</i> ) | L           |
| N    | Surveillance, National  | L           |
| C    | Concentration Point Targeted Surveillance   | L           |

## Order Group for Subtyping PCR

Subtyping PCR is often treated as a single “test” because it is run in a single tube. However, the “test” actually produces four distinct “results”; one for each of the H1, H3, N1, and N2 gene segments. For simple reading by humans, this is usually reported as a simple string, “H1 N1” or “H1 N1 and N2” or even “Negative” if none of the four reached threshold. For Marvin the Manic Depressed Robot, and other computers we need to explicitly spell out the four results. And, while the Ct values are not important to clinicians or epidemiologists, we want to save that information for future reference.

To meet all these needs, the result of subtyping PCR is reported as one ORDER group with the order-only LOINC code X0003-4 “IAV-S Subtyping PCR Sub-protocol” with four RESULT groups. Each RESULT group has one OBX with the LOINC code for PCR on that specific H or N. The answer in OBX.5 is the Ct value reported for the specific H or N. The “POS/NEG” values go in the interpretation field, OBX.8.

If necessary, it is *not wrong* to repeat the ORDER group four times with the OBR.4 LOINC code X0003-4 and one RESULT with the LOINC code for the specific H or N PCR. Some LIMS may be set up to only send one-to-one order to result. Receiving systems can re-group these either when storing or retrieving the information. This is a clumsy, but technically acceptable way of sending this information. Even further removed from reality but workable is the option of sending each result with the result LOINC in both OBR.4 and OBX.3 as if the four individual PCRs had each been ordered separately.

### Sequencing Results

Historically, when clinicians ordered genetic sequencing, they were interested in the presence or absence of a specific mutation or the identification of a specific haplotype. This was true both for host genome sequencing and for pathogen sequencing. The idea that they would ask for sequencing and expect to get back an actual series of bases “AGCTGGATACA...” was a foreign concept. But in the IAV-S program that is exactly what we need in order to track the evolution of different Influenza A strains. After considering various options the NAHLN elected to have laboratories submit the sequences to GenBank, which they were going to have to do anyway, and then consider the GenBank accession to be the result.<sup>43</sup> The information this provides is, in fact, the complete sequence along with the GenBank metadata so the test really is providing the full genetic sequence requested. Plus, the GenBank metadata includes much of the information an epidemiologist or clinician would want such as the precise strain name for the virus. With the accession number or unique specimen ID, it is a simple matter to look this up on GenBank either through the web site, the Influenza Research Database, or in other software by calling on the eUtils web services.

Because the specimens are barcoded, and because GenBank can be searched by specimen ID, it is really not necessary to message sequence results. To close the loop we message the GenBank accession number in OBX.5 either as a simple string (current implementation) or as EI.1 in a more correct rendering.

### Porcine Enteric Coronavirus Testing

Porcine Epidemic Diarrhea (PED) along with the related but distinct species of coronavirus Porcine Delta Coronavirus (PDCoV) are a good example of a rapidly emerging disease of national importance with an important laboratory component. The data requirements have a great number of similarities to all the other NAHLN testing

---

<sup>43</sup> The use of GenBank is beyond the scope of this manual but can be found in the *Encyclopedia Galactica* or on the GenBank web site at <http://www.ncbi.nlm.nih.gov/Sequin/acc.html>

programs. There are just a few wrinkles, as there almost always will be. In this section we will look at the various tests available and how we may be able to handle some of the peripheral epidemiologic data required by this testing program.

### **Conventional vs. Real-time PCR**

As testing for PED and related viruses has been rapidly developed, we have seen the introduction of both conventional and real-time polymerase chain reaction (PCR) tests. For purposes of LOINC coding, nested PCR is a conventional (qualitative) PCR. In order to properly indicate the test run, we need to use the correct LOINC codes (and sometimes method OIDs, though those have yet to be used). Sometimes only the interpretation of a real-time PCR test is sent in the message. However, if a Ct value exists even if not reported out, in veterinary medicine we use the quantitative LOINC code. Only if the test itself generates only positive or negative do we use the ordinal LOINC. But it is wrong to report positive or negative as the result of a quantitative test. What we do is send a “flavor of null” as the result and “POS” or “NEG” as the interpretation in OBX.8. To indicate that we know the actual result is quantitative but not present in the message, OBX.5 should contain the code “NP” meaning “Not Present”. Of course, it may be easiest and most correct to send the Ct value along with the interpretation.

The LOINC codes for these tests are on the terminology services site.

### **Age Categories**

The idea behind NAHLN messaging is for most of the data in each message to apply to any, or at least most, programs, kinds of testing, etc. But real-life sometimes intrudes and submitters will send information in a format specifically requested by a program that may or may not fit well in the LIMS' schema or the message. One such data element is age category for PED testing. The epidemiologists in this case want to see submissions broken down into age categories, suckling, nursery, grower/finisher, and sow/boar. The PID segment has a place for date of birth, but not for this kind of category. We can add the category as an observation following the PID segment. (See the earlier section on Age Group.) But most labs have never needed to implement this feature of the message. So how can we work around this?

The ideal solution is to collect date of birth and report that in PID. The folks creating the reports can then use the definitions of the groups however they want to create age groups. If the submitter sends a collection date and an age on that date, you can calculate a date of birth. But what if the submitters only send groups as defined by the program?

## The Hitchhiker's Guide to NAHLN Messaging

The best solution if you only receive the age group is to report the age group using SNOMED.

| Age Group       | SNOMED Code     | SNOMED Text  |
|-----------------|-----------------|--|
| suckling        | 28911000009108  | Suckling pig for meat production (organism)        |
| nursery         | 28901000009106  | Starting pig for meat production (organism)        |
| grower/finisher | 356661000009103 | Grower/finisher pig for meat production (organism) |
| sow/boar        | 28811000009104  | Adult pig for meat production (organism)           |

```

<OPU_R25.PATIENT>
  <PID>
  ...
  </PID>
  <OPU_R25.PATIENT_OBSERVATION>
    <OBX>
      <OBX.2>ST</OBX.2>
      <OBX.3>
        <CWE.1>46251-5</CWE.1>
        <CWE.2>Age Group</CWE.2>
        <CWE.3>LN</CWE.3>
      </OBX.3>
      <OBX.5>
        <CWE.1>28911000009108</CWE.1>
        <CWE.2>Suckling pig for meat production (organism)</CWE.2>
        <CWE.3>SCT</CWE.3>
      </OBX.5>
      <OBX.11>F</OBX.11>
    </OBX>
  </OPU_R25.PATIENT_OBSERVATION>
</OPU_R25.PATIENT>

```

But what if your system does not support sending PATIENT\_OBSERVATION?

One solution that has been used is to pick a midpoint age and calculate an estimated DOB from that and the collection date (or receipt or test date if that is all that is available). You *could* be a little more correct by omitting the day portion of the DOB for those older than suckling and just report birth month but most will just use the full midpoint date.

| MAPPING         |   | Time Period                   |
|-----------------|---|-------------------------------|
| Age Group       | Definition                                    | Subtract from Collection Date |
| suckling        | suckling (<1 month or still on sow)           | 2 weeks                       |
| nursery         | nursery (1 month up to 3 months)              | 2 months                      |
| grower/finisher | grower/finisher (three months up to 8 months) | 6 months                      |
| sow/boar        | sow/boar (8 months or older)                  | 10 months                     |

So, if a submission came in with a collection date of May 15, 2014 and an age group of “nursery” *and* you have no way to send the age group itself, you would calculate a birth date of March 15 by subtracting two months from May 15. You would then send it as <PID.7>20140315</PID.7>.



Any conversion between age group and date of birth loses information. We *never* want to do this conversion before the data are stored in the LMS system. Conversion to categories is a different issue for analysis on the other end. That is the right place to do any aggregation. There it can be directed and controlled by the epidemiologists using the information. The only reason to do a conversion on the sending end would be that the system cannot collect, store, or send the data as submitted.

Order of Preference for Age:

1. Receive specific date of birth from submitter and send date of birth in PID.7
2. Receive age from submitter and calculate date of birth to send in PID.7
3. Receive age group from submitter and send as a patient observation as above
4. Receive age group from submitter and calculate date of birth using logic above to send in PID.7 (You should only do this if the sending system does not support patient observation.)

### Unique Specimen Identifiers

Because most of these submissions come directly from industry veterinarians rather than program specimen collection activities they will not come in with globally unique specimen IDs. Instead labs will need to report whatever ID did come on the submission along with a unique ID assigned by the lab. This unique ID may have to be constructed from the accession number plus a specimen sequence, etc., if the LIMS does not make a system-unique ID available. The placer ID's assigning authority may then be omitted or a text description "Farm ID" or similar if provided by the LIMS can go in EI.2 just for human readability in knowing the type of ID. The important unique ID is then the filler ID (EIP.2) and it needs the lab's OID (or Lab's LIMS OID or Lab's specimen ID OID as appropriate) in EI.3 and "ISO" in EI.4.

```
<SPM.2>
  <EIP.1>
    <EI.1>123</EI.1>
  </EIP.1>
  <EIP.2>
    <EI.1>C14001234-1</EI.1>
    <EI.3>2.16.840.1.113883.3.5.1.9</EI.3>
    <EI.4>ISO</EI.4>
  </EIP.2>
</SPM.2>
```

Because the placer specimen ID is used in human-readable presentations of the messages, it is important that this field include *something*. In cases where no identification was provided or recorded from the submitter, the sending system should repeat the Filler ID as if it was the placer ID. While this sounds wrong, the lab's OID in EI.3 will make clear that the "Placer ID" was actually provided by the laboratory as the specimen was received.

```
<SPM.2>
  <EIP.1>
    <EI.1>C14001234-1</EI.1>
    <EI.3>2.16.840.1.113883.3.5.1.9</EI.3>
    <EI.4>ISO</EI.4>
  </EIP.1>
  <EIP.2>
    <EI.1>C14001234-1</EI.1>
    <EI.3>2.16.840.1.113883.3.5.1.9</EI.3>
    <EI.4>ISO</EI.4>
  </EIP.2>
</SPM.2>
```

## Environmental Samples

Environmental samples pose an interesting specimen type issue. The specimen actually submitted is usually a swab, sponge, etc. that was used to collect whatever was on the surface of the environmental place or thing being tested. What we really care about is what was swabbed, sponged, etc. That goes in the specimen source field. For PED the only real distinction that is critical to the program (and to labs getting paid!) is whether the specimen came from a farm environment or some other place like a truck wash, market, etc. So, this becomes reasonably simple when just considering the PED/PDCoV program as the example shows.

```
<SPM.4>
  <CWE.1>419695002</CWE.1>
  <CWE.2>Environmental swab (specimen)</CWE.2>
  <CWE.3>SCT</CWE.3>
</SPM.4>
<SPM.6>
  <CWE.1>224816003</CWE.1>
  <CWE.2>Farming environment (environment)</CWE.2>
  <CWE.3>SCT</CWE.3>
OR
</SPM.6><SPM.6>
  <CWE.1>356671000009107</CWE.1>
  <CWE.2>Non-farming environment (environment)</CWE.2>
  <CWE.3>SCT</CWE.3>
</SPM.6>
```

## ***Bovine Spongiform Encephalopathy (BSE) (Draft)***<sup>44</sup>

In *The Hitchhiker's Guide to the Galaxy*, the spaceship *Heart of Gold* is capable of producing “something almost but not completely unlike tea.” In the Bovine Spongiform Encephalopathy (BSE) testing program, specimen collectors seem to be pretty good at collecting “something almost but not completely unlike obex.” Because of the sensitive nature of BSE testing there are a number of very strict procedural rules—including how to handle less than ideal specimens—that must be followed both in the conduct of the testing and in reporting results. These rules must carry through to the automated results reporting.

<sup>44</sup> As of this writing the BSE program is still using a non-standards-based XML format for results. This discussion is based on the planned but never implemented standard result message.

When labs receive a specimen that appears to be brainstem but is not, or cannot be determined to be, at the level of the obex, they go ahead and test the specimen, but report it out as, “Not detected, not obex.” This “result” is actually two distinct bits of information. First, “Not detected” tells us that the assay was performed and did not detect the abnormal prion protein. But we also learn that the specimen was not the desired specific part of the brainstem. These tests are treated differently when doing national epidemiology because the negative predictive value is not as strong as those tests performed on the ideal specimen. So, how do we message this?

Computers demand that things go in nice neat boxes. “One bit of information to a box please!” So, “Not detected, not obex” is not a valid result. Instead, we need to say, “Not detected” but “this was not the ideal specimen” because it was “brainstem” but not necessarily “brainstem at the level of the obex.” We have a whole paragraph, not just one word. And the message has places for all the pieces of our paragraph. The result goes in the Result OBX.5 “Observation Value.” The details of the specimen go in the SPM segment. The type “Specimen from the brain” (119398007) goes in SPM.4 “Specimen Type” and the location “Brainstem structure” (15926001) goes in SPM.8 “Specimen Source Site” and the notation that this is an acceptable but not ideal specimen goes in SPM.23 “Specimen Appropriateness.” (A) If the reason we can't say it is obex is that the quality is poor, we can indicate that in SPM.22 “Specimen Quality” and/or SPM.24 “Specimen Condition.” If we get the correct specimen, things look the same except SPM.8 contains “Brain stem at the obex” (741000009106) and SPM.23 labels the specimen appropriateness as “Preferred” (P) and condition and quality are labeled “good” or “excellent.”

All this attention to detail is not simply an academic, “make work,” exercise. Having the data in standardized, unambiguous locations makes all sorts of retrospective analysis possible. Suppose NVSL comes out with a new training video on how to collect and process obex specimens. A few months later they want to know if it had an impact on improving specimen quality. With our well-defined structure they can look at quality, and appropriateness in as much detail as necessary. Likewise, an epidemiologist can easily summarize all results or only those performed on good quality, appropriate specimens.

Once we go live with HL7 messaging of BSE, much of the discussion, especially the use of LOINC and SNOMED codes from the Scrapie section that follows will apply.

### **Scrapie**

Like BSE, the On-farm Scrapie Surveillance and Regulatory Scrapie Slaughter Surveillance (RSSS) present some unique challenges for results messaging. In this chapter, we will discuss scrapie messaging in general. Some of the material will only apply to national reference laboratory testing, while other parts apply to member diagnostic laboratories as well. As with other formal NAHLN testing programs, see the Program-Specific *LMS Scrapie Messaging Program Guide* available on the terminology service website.

Testing for scrapie involves several different tissues collected at slaughter or necropsy. For efficiency these specimens are placed together in a common container of formalin. The entire container gets a single specimen identifier. This results in an SPM.2, "Specimen ID" with a fairly complicated structure. Each specimen gets its own SPM segment. The Specimen ID is actually a *pair* of identifiers; one assigned by the collector ("Placer Assigned Identifier") and another assigned by the lab ("Filler Assigned Identifier"). Each of these has its Assigning Authority OID. The Specimen ID must be unique. Normally, both placer and filler IDs are unique. In this case however, the placer ID is repeated for each specimen in the container. They are distinguished for the placer by the specimen type, "The third eyelid specimen from container 12345." The filler ID is generated by the lab and is unique to each individual tissue in the container. The filler ID serves as the globally unique identifier for each specimen. Sometimes collectors send what they think is one tissue, but histological examination shows to be something else. If the actual tissue is something inappropriate for testing, this can be indicated in SPM.23, "Specimen Appropriateness." Note that the message does not contain a field for "what the collector thought they sent!" It would not be possible for the lab to consistently determine this from the scrapie submission process.

The process of testing for Scrapie and other Transmissible Spongiform Encephalopathies (TSEs) is an interesting exercise in "what exactly are we looking for?" In the dark ages when the *Encyclopedia Galactica* was still considered the definitive source for such information, we thought that TSEs were caused by a virus. It was later found that the actual etiologic agent was a malformed protein called a "prion." Prions exist normally in brain and some other tissues. When abnormally folded they take on two special characteristics. In this abnormal configuration they can induce other normal prions to fold into the abnormal shape. This is what makes them communicable. The abnormal shape also makes them resistant to being broken down by protease enzymes. This is involved in causing disease and is also used in diagnosis.

The most basic tests for TSEs simply detect the presence of prion protein that is resistant to protease in vitro. In this sense, the tests do not look for Scrapie or BSE but simply "abnormal prion protein." Later tests used immune assays to detect the abnormally folded protein and were tuned and labeled for use in one species but still detected "abnormal prion protein." These tests are still the main ones used for TSE screening, including scrapie testing. Finally, even more specific tests were developed that could detect differences between the types and subtypes of TSEs. This is important for differentiating atypical scrapie, Nor98, from typical scrapie (and the H and L atypical BSE variants from typical BSE). These tests detect the "type" of abnormal prion protein. Finally, microscopic techniques that do not involve a prion protein specific immune stain do not detect the abnormal prion protein at all, but rather the anatomic pathologic signs of the disease process itself. Correctly coding these tests is important because it indicates the level of specificity available to the test. Such information could become vital as additional atypical strains are discovered for example.

## The Hitchhiker's Guide to NAHLN Messaging

When a test detects the type of abnormal prion protein or TSE, the result (OBX.5) is a SNOMED code for either the agent or the disease. It may seem like splitting hairs, and an analyst could probably figure out what was meant if the agent was used for a histopathologic result or the disorder used for a test that really detects the agent, it is best to get these right. And for the most part, NAHLN labs will be doing the screening tests that return a positive/negative result. It will be the reference labs doing the follow-up subtyping that will deal with the codes for TSE subtypes.

Below are tables of the LOINC codes used in OBR.4 and OBX.3 (Order and Result) and the SNOMED codes for appropriate answers.

The LOINC code selection is based on the type of test—IHC, western blot, or ELISA—the general type of specimen—brain vs. other tissue types—and whether the test would return simply “positive” or would identify the specific type of TSE identified such as “Atypical Scrapie Nor98” for a confirmation scrapie test. Because “tissue” can include “brain,” and because the specimen type is always included in NAHLN result messages, the new confirmation and typing codes only come as “tissue.” It is *not incorrect* to use the “tissue” codes with brain specimens in any case. It *is* incorrect to use the “brain” codes with lymphoid tissue, etc.

### TSE LOINC Codes used for Scrapie and other TSEs

| Test                                 | LOINC Code | LOINC Name                    | Result Value Table                             |
|--------------------------------------|------------|-------------------------------|--|
| Western Blot                         | 39320-7    | Abn Prion Prot Brain Ql IB    | Qualitative                                    |
| IHC                                  | 39020-3    | Abn Prion Prot Brain Ql ImStn | Qualitative                                    |
| ELISA                                | 37425-6    | Abn Prion Prot Brain Ql EIA   | Qualitative                                    |
| Western Blot                         | 39320-7    | Prion Protein Tissue Ql IB    | Qualitative                                    |
| IHC                                  | 39020-3    | Prion Protein Tissue Ql ImStn | Qualitative                                    |
| ELISA                                | 39021-1    | Prion Protein Tissue Ql ImStn | Qualitative                                    |
| Histopath                            | 38275-4    | Prion Prot Brain Ql Micro     | Qualitative                                    |
| Confirmation and Typing IHC          | 74035-7    | Prion Prot Tissue Prid ImStn  | Presence or Identity of Abnormal Prion Protein |
| Confirmation and Typing Western Blot | 74036-5    | Prion Prot Tissue Prid IB     | Presence or Identity of Abnormal Prion Protein |
| Confirmation and Typing Histopath    | 74037-3    | TSE Prid Brain by Microscopy  | Presence or Identity of TSE                    |

### Qualitative SNOMED Results Codes

| SNOMED Code | SNOMED Text          |
|-------------|----------------------|
| 281297005   | Analyte not detected |

|                 |                      |
|-----------------|----------------------|
| 260385009       | Negative             |
| 280416009       | Indeterminate result |
| 10828004        | Positive             |
| 359951000009103 | Analyte detected     |

Notice that there are two very slightly different SNOMED codes for “negative.” Both “Analyte not detected” and “Negative” mean that the test did not detect the agent identified by the LOINC code. “Analyte not detected” allows a slightly less emphatic “no” if needed for various reasons.

When it comes to the answers on confirmation and typing tests, we really split hairs. There is a difference between identifying the type of abnormal prion protein directly as we do with IHC or western blot tests and identifying the pathology itself as is done with ordinary histopathology.

#### Presence or Identity of *Abnormal Prion Protein* SNOMED Result Codes

| SNOMED Code     | SNOMED Text   |
|-----------------|---|
| 281297005       | Analyte not detected  |
| 333241000009108 | Scrapie encephalopathy agent (organism)                             |
| 333281000009103 | Atypical scrapie encephalopathy agent - Nor98 (organism)            |
| 68638009        | Bovine spongiform encephalopathy agent (organism)                   |
| 333251000009106 | Atypical bovine spongiform encephalopathy agent (organism)          |
| 333261000009109 | Atypical bovine spongiform encephalopathy agent - H type (organism) |
| 333271000009100 | Atypical bovine spongiform encephalopathy agent - L type (organism) |
| 361000009131    | Chronic wasting disease encephalopathy agent (organism)             |

#### Presence or Identity of *TSE Pathology* SNOMED Result Codes

| SNOMED Code     | SNOMED Text  |
|-----------------|--|
| 260385009       | Negative   |
| 35103004        | Scrapie (disorder)                                   |
| 333361000009103 | Atypical scrapie encephalopathy Nor98 (disorder)     |
| 52869003        | Bovine spongiform encephalopathy (disorder)          |
| 333301000009102 | Atypical bovine spongiform encephalopathy (disorder) |
| 333291000009101 | Chronic wasting disease (disorder)                   |

### Specimens in Scrapie

Specimen types and the way they are handled are somewhat different in the scrapie program. Specimens may be submitted as multiple tissues in one container of fixative as they are sometimes for diagnostic histopathology. This works for human consumption but breaks down in trying to follow the pattern in which the specimen identifier is the key to linking results back to orders. In scrapie, the specimen “pool” identifier must be combined with the specimen type to uniquely identify a specimen. There has been

significant gnashing of teeth over cases in which the submitter misidentifies the tissue in one or more specimens.

For messaging, the specimen should always be identified by the label on the “tube” as the placer specimen identifier and the scrapie program OID as the placer ID assigning authority. The filler specimen identifier is then needed to provide a truly unique identifier. The specimen type reported should be what was actually identified in the submitted specimen regardless of what the submitter called it.

### Scrapie Specimen Types

| SNOMED Code     | SNOMED Text   |
|-----------------|---|
| 258589002       | Lymph node sample (specimen)                        |
| 356871000009108 | Mesenteric lymph node tissue specimen (specimen)    |
| 4251000009104   | Retropharyngeal lymph node tissue sample (specimen) |
| 811000009103    | Tonsil  |
| 356561000009102 | Obex Only   |
| 356631000009105 | Brainstem not Obex                                  |
| 356621000009108 | Brainstem including Obex                            |
| 356621000009108 | Entire Brainstem including Obex                     |

### Non-result and Qualified-result Results in Scrapie

Up to now, everything in this chapter is more-or-less the same as other NAHLN messaging. Non-results, however, are almost but not entirely unlike any other messaging. This is a result message after all, but these are either not results at all or results that don't really count in the national surveillance system. Nevertheless, the program needs to keep track of these cases that come about for a number of reasons.

Sometimes the wrong specimens are submitted as described above. In other cases specimen condition may prevent testing. Or additional specimens may not be needed once the most preferred specimens have been tested. Finally, sometimes it isn't until a specimen is tested—usually by immunohistochemistry—that the technician recognizes something such as inadequate lymphoid follicle tissue to make a good interpretation. Each of these scenarios bend but don't completely break the usual NAHLN messaging logic.

The various issues are carried in different places in the message. This bothers some who are used to the “result” string on the client report including many different individual facts. In messaging, each fact has a specific field. Specimen type is in Specimen Type (SPM.4), Specimen Condition in Specimen Condition (SPM.24), and actual observation (OBX.5) distinct from the interpretation of the observation (OBX.8). Other fields may carry further distinctions. The challenge on both the sending and receiving ends is to map to and from user-friendly result strings to computer-friendly distinct fields. The following table maps out a number of common non-results and qualified-results.

## The Hitchhiker's Guide to NAHLN Messaging

| Scrapie Program Qualifier | Scrapie Program Definition                               | SPM.4,6                      | SPM.2 1 | SPM.2 2   | SPM.2 3      | SPM.24 | ORC.5 | OBX.5  | OBX.8 | OBX.11 |
|---------------------------|--|------------------------------|---------|-----------|--------------|--------|-------|--|-------|--------|
| Insufficient Follicles    | Insufficient Lymphoid Follicles                          | Actual Specimen              | SC      | E or Omit | A or I       | ISF    |       | Observed Result                                  | ILF   | F      |
| Location                  | Incorrect Sample Location                                | Actual Specimen              | IS      | E or Omit | A or I       |        |       | Observed Result                                  | ISL   | F      |
| Repeat                    | Unclear result for any other reason requiring retesting. | Actual Specimen              | Omit    |           |              |        | CA    | Omit   | SCR   | X      |
| Suspect                   |  | Actual Specimen              |         |           | A            |        |       | Observed Result                                  | SCS   |        |
| Unsuitable                | sample unsuitable for this assay                         | Actual Specimen              | IS      | E or Omit | I            | Omit   | CA    | Omit   | SCU   | X      |
| Unacceptable              | specimen unsatisfactory for evaluation                   | Ordered Specimen             | SC      |           |              | Why?   | CA    | 125154007 specimen unsatisfactory for evaluation |       | X      |
|                           | Autolysis  | Ordered Specimen             | SC      | P         | Omit or P    | AUT    | CA    | Omit   |       | X      |
|                           | Broken Container   | Ordered Specimen             | RB      |           |              |        | CA    | Omit   |       | X      |
|                           | No sample ID   | Ordered Specimen             | RI      |           |              |        | CA    | Omit   |       | X      |
| No Sample Received        | no sample received                                       | 281266006 no sample received | QS      |           |              | SNR    | CA    | Omit   |       | X      |
| Not Tested                | test not done on specimen because not needed             | Actual Specimen              | Omit    | E or Omit | P, A or Omit | Omit   | CA    | 373121007 Test not done                          |       | X      |

### ***Non-Program Initiated Testing***

Up to now just about everything we have discussed was based on specific national surveillance program testing. In general these programs have additional data streams that often make it possible to send very minimal “demographics” along with the result. For those program tests, the globally-unique specimen barcode is sufficient to identify the subject of the testing included in the report. All that changes with testing that originates with the submitting veterinarian.

When the NAHLN repository was replaced by the Laboratory Messaging System (LMS) the system gained the ability to receive a much wider range of tests. These tests may or may not eventually have some information hand-entered into systems such as the Surveillance Collaboration System (SCS) from forms such as the VS 4-54, but the result message must be assumed to be the entire “source of truth” for these tests. The simplified result messages that many labs have grown accustomed to will not suffice for these more general test results. This section gives a basic overview of the additional requirements.



## Premises Role

To have any value at the national level test results must contain at least minimal information about *where* the test specimens were collected. The goal is to have every collection premises identified by a globally unique federal Premises Identification Number (PIN) or state-issued Location Identification (LID). This information goes in ROL.13/PL10/EI.1. Until such identifiers are universally available, the next best option is to include complete physical address ("911-address") in ROL.11 Address. If a PIN or LID is provided in PL.10, the physical address may be omitted, but it is good practice to include it in any case. This will provide an effective cross-reference.

At an absolute minimum, ROL.11/XAD.4 State Abbreviation is essential. Access to the LMS repository will very likely be based at least in part on the state in which the specimen was collected.

An absolutely minimal PREM ROL would look something like this:

```
<ROL>
  <ROL.2>UC</ROL.2>
  <ROL.3>
    <!--The Role being played by the person or organization represented by this segment-->
    <CWE.1>PREM</CWE.1>
    <CWE.2>Collection Premises</CWE.2>
    <CWE.3>HL70443</CWE.3>
  </ROL.3>
  <!--The main purpose of this segment being identification of people in roles makes
  ROL.3 required. Here it is simply NA, not applicable. -->
  <ROL.4>
    <XCN.2>
      <FN.1>NA</FN.1>
    </XCN.2>
  </ROL.4>
  <!--The one absolutely essential element is the collection state! -->
  <ROL.11>SC</ROL.11>
</ROL>
```

## Submitter Role

For some regulatory testing it is important that we know the accredited veterinarian who submitted the samples for testing. Another ROL segment handles this. This one is the Submitter (SUB) role. The best and most authoritative identifier for submitters of regulatory tests is the national veterinary accreditation program (NVAP) number. This goes in ROL.1, Role Instance Identifier. The assigning authority OID in EI.3 is 2.16.840.1.113883.3.5.6.3, "AAVLD\_External\_Code\_Systems/National Veterinary Accreditation Program".

```

<ROL>
  <ROL.1>
    <!--Identifier of the entity in this role. Here accredited veterinarian submitter. -->
    <EI.1>002345</EI.1>
    <EI.3>2.16.840.1.113883.3.5.6.3</EI.3>
    <EI.4>ISO</EI.4>
  </ROL.1>
  <ROL.2>UC</ROL.2>
  <ROL.3>
    <!--The Role being played by the person or organization represented by this segment-->
    <CWE.1>SUB</CWE.1>
    <CWE.2>Submitter</CWE.2>
    <CWE.3>HL70443</CWE.3>
  </ROL.3>
  <ROL.4>
    <XCN.2>
      <FN.1>Roberts</FN.1>
    </XCN.2>
    <XCN.3>James</XCN.2>
  </ROL.4>
</ROL>

```

To date, no one has begun messaging the accreditation number but it is a good example of an identifier for the person *in the role*. Otherwise, ROL.1 is normally omitted.

## PATIENT Group

When submitting non-program initiated animal test results, the PATIENT group is no longer optional.

Perhaps the most critical element for regulatory testing is patient identification. The PID.3 Patient Identifier List can contain both official and unofficial identification. Official identification under the Animal Disease Traceability program has the OID “2.16.840.1.113883.3.5.6.1.2” supplied in CX.4. Farm tags, etc., are simply supplied in CX.1 with no CX.4 included. [Program Question: Is there a reason to distinguish between AINs and NUES, Scrapie, and other “Official Animal Identifiers” or should we call these all simply “ADT Official Identifiers”? Most current systems simply call them all “official” if they make any distinction at all.]

At least basic signalment information is required. Patient name is required by the standard from its human origins. This can be set to a null flavor such as NA, NI, or Not Provided as discussed earlier if no name is available. Patient sex (PID.8) and taxonomy (PID.35) are essential as is Production Type in most cases. (*Very* rare cases may have unknown sex but as a general rule, this field must be populated.) Some indication of age is usually important. The HL7 date type allows less than exact precision. Year of birth can be expressed as simply YYYY (eg. “2008”).

If it is impossible to express patient age in terms of date of birth of any precision, it is possible to send just an age category such as “Adult.” This is technically an age group based on an ordinal scale but the scale can be local. To do so, a PATIENT\_OBSERVATION group follows the PID and includes a single OBX segment with LOINC code 46251-5 “Age Group”. OBX.5 is a ST data type the value of which is supposed to be an ordinal value. This should eventually be modeled as a CWE with the Age code in CWE.1, a description (likely the same word) in OBX.2 and “L” in OBX.3 or

SNOMED Age Group as used with swine. Depending on usage, the ordinal values for this application may be standardized at some future date in which case, a code system OID will be provided along with codes and descriptions. We first need to standardize age categories for each important species. (See also: *Age Categories* in the *Swine Influenza and Porcine Enteric Coronavirus* chapters.)

In the case of natural death, the patient death indicator and date of death (PID.30 and PID.29) become important. PID.30 is set to “Y” and the date of death to an HL7 formatted date/time to as much precision as is known. This may or may not include time.

```
<OPU_R25.PATIENT>
  <PID>
    <PID.3>
      <!--Official ID has an assigning authority. -->
      <CX.1>840000000134643</CX.1>
      <CX.4>
        <HD.2>2.16.840.1.113883.3.5.6.1.2</HD.2>
        <HD.3>ISO</HD.3>
      </CX.4>
    </PID.3>
    <PID.3>
      <!--Farm tag in this case does not. -->
      <CX.1>Y_1234</CX.1>
    </PID.3>
    <PID.5>
      <XPN.1>
        <FN.1>NOT PROVIDED</FN.1>
      </XPN.1>
    </PID.5>
    <PID.7>2008</PID.7>
    <PID.8>F</PID.8>
    <!--Natural death on Oct 29, 2013. -->
    <PID.29>20131029</PID.29>
    <PID.30>Y</PID.30>
    <PID.35>
      <CWE.1>26105007</CWE.1>
      <CWE.2>Holstein-Friesian cattle breed (organism)</CWE.2>
      <CWE.3>SCT</CWE.3>
    </PID.35>
    <PID.38>
      <CWE.1>DA</CWE.1>
      <CWE.2>Dairy</CWE.2>
      <CWE.3>HL70429</CWE.3>
    </PID.38>
```

## Specimen ID

Many non-program initiated submissions do not arrive with globally unique specimen identifiers. In these cases all you can send for the Placer Specimen Identifier is what was actually on the tubes, bags, etc. These have no assigning authority OID. The identifier is *not* guaranteed to be unique. In these cases you may only populate SPM.2/EIP.1/EI.1. It is especially important in these cases to correctly populate the Filler Specimen Identifier with an identifier created by your laboratory that *will* uniquely identify this specimen. The OID for that identifier is your lab's OID or the OID created by your lab for specimen identifiers if you have subdivided your OIDs.

```
<SPM>
  <SPM.2>
    <EIP.1>
      <!--Label on tube. Not unique. -->
      <EI.1>1</EI.1>
    </EI.1>
    <EIP.2>
      <!--Lab assigned ID. Must be unique. Could be database key. -->
      <EI.1>C13001234/12345</EI.1>
      <EI.3>2.16.840.1.113883.3.5.1.9.10</EI.3>
      <EI.4>ISO</EI.4>
    </EI.2>
  </SPM.2>
  ...
</SPM>
```

### Purpose of Tests

The NAHLN list of purpose for tests was very limited. The only value used so far was N for National Surveillance. This value set may need to expand to allow inclusion of information such as “Brucellosis whole herd testing” or “Case trace follow-up.” This remains work to be accomplished. See individual program messaging guides for details.

### Conclusion

You have reached the end of this version of the *Hitchhiker's Guide to NAHLN messaging*. This is not the end however. Just as with the original *Hitchhiker's Guide to the Galaxy*, this is intended to be a living document. It will be updated regularly as we gain experience with using the NAHLN messaging system, as new message types are introduced, and as you the readers suggest areas where the guide could be more helpful. (Comments, suggestions, complaints, etc., may be sent by email to the author at [mmarti5@clemson.edu](mailto:mmarti5@clemson.edu).)

### Acknowledgements

In addition to contributions by all of the members of the NAHLN IT Committee who have read and commented on this living document a few have made significant contributions of content and editorial assistance. Jim Case, the lead designer of the NAHLN HL7 messaging and author of the *Implementation Guide(s)* provided numerous examples and technical editing to ensure correct description of message usage and interpretation. Julie Green provided coverage of the terminology services and extensive editing of all the terminology discussion. Jay Kammerzell, Shane Linke, and Scott Ross provided material on the Rhapsody interface engine. And special thanks go to Jay Weidner for repeated and detailed copy editing. Anyone who has tried to write a document like this knows how hard it is to get anyone to do detailed proof-reading, and how difficult a task that proofing is. Jay has done excellent reviews not once, but several times as the document has evolved. Sally Dubose at Clemson LPH provided another very thorough review. Finally, all those who have posted questions on the forum have contributed by helping to focus this document on the real challenges faced by labs working to set up messaging. The “chief scribe” has undoubtedly added errors faster

than Jay and the rest could find and correct them, but all these contributions have helped immeasurably.

To all of the above, please accept my heartfelt “AA” Acknowledgement.

## **Notices**

"This material contains content from LOINC (<http://loinc.org>). LOINC is copyright © 1995-2020, Regenstrief Institute, Inc. and the Logical Observation Identifiers Names and Codes (LOINC) Committee and is available at no cost under the license at <http://loinc.org/license>. LOINC® is a registered United States trademark of Regenstrief Institute, Inc."

## **Future Topics**

### **Additional Messaging Specific Disease Control Program Tests**

#### **Messaging Specific Types of Tests**

- Polymerase Chain Reaction**

- Enzyme-Linked Immunosorbent Assay**

- Virus Isolation**

- Serology**

#### **Approaches to messaging**

- Single test result messages**

- Multiple test result messages**

#### **Errors**

- What kind of errors are there?**

- How do I fix them?**

- Correcting data**

- Resending data**

#### **Future expansion**

- Support for order messages for referral testing and use outside of NAHLN**

- Lab-to-lab transmission**

- Expanded disease coverage**

- Interfacing with other networks**