

# HL7 Conformance Checking with Corepoint Integration Engine™

## Overview

Although the HL7 2.X messaging standard is the most widely used standard in the United States for the exchange of clinical patient data, it varies greatly in how it is implemented by each medical device and application. Consequently, it is often called the “non-standard standard”. The purpose of HL7 2.X is to provide a framework for negotiation so that each healthcare interface is closer to 20% custom rather than 100%.

The variances in implementation lead to different message formats among healthcare vendors and external providers. In order to communicate effectively between systems with different message formats, you must determine where the messages are incompatible and make changes to at least one, and potentially both, of the interfaces that are accepting or sending messages.

Gap analysis or *conformance checking for HL7 messages*, is a logical process used to determine whether a message from one particular medical device or application is compatible to the standard HL7 messaging format, or a custom format, adopted by another device or application.

## Why Does Nonconformance Occur?

Nonconformance occurs for two main reasons:

1. An application team utilizes the flexibility of the HL7 2.X message standard to meet their system’s unique requirements. This occurs primarily in the areas of cardinality and the HL7 version that is used.
2. The HL7 messaging standard can be complex and sometimes is easily misinterpreted.

## Cardinality

Cardinality, another cause of nonconformance, represents the minimum and maximum of number of values that could exist inside a given element of the message. The HL7 messaging standard allows segments and fields to be either optional or required, and either singleton (non-repeating) or repeating.

This same functionality exists for sub-components as well in 2.5. For example, the Patient Address field (PID-11) can be optional and repeating. One development team may implement this exactly as the standard says and allow zero to infinite patient addresses. Another development team may require one patient address but not allow more than one. Messages from these two systems would not be conformant with each other.

HL7 2.X is designed to be backward compatible to work with systems using different versions. However, there are a few instances where problems occur, such as when message triggers vary from version to version, for example MDM-TO2 exists in 2.3 but not in 2.2.

An application development team or implementation team may adjust the HL7 messaging standard to better support their application or system. Sometimes these adjustments make the message format used by that application or system noncompliant with the HL7 standard. For example, an application may have a database that only has one field for a patient alias and therefore the application only allows one patient alias to be entered in the GUI. The HL7 standard says that the patient alias field can repeat, so it is conceivable that a message received through an HL7 interface would have more than one patient alias.

Finally, while development teams are experts in their specialized area, such as a lab or pharmacy, that expertise does not necessarily transfer to HL7 messaging. The HL7 messaging standard, while extremely flexible, is also complex. Additionally, interfacing with other applications may not be the highest priority on the list when creating a clinical application. By the time interfacing is approached, decisions regarding how the application will work may have already been made. This could result in any number of messages from the application to lack HL7 compliance.

Like most things regarding HL7 2.X, these differences need to be negotiated between the external healthcare vendors trying to exchange data.

---

## How Do You Check for Conformance?

As an analyst trying to ensure correct communication between medical devices and applications, you will have to look at all the possible predictable and unpredictable causes for nonconformance. Where do you start?

- First, determine how closely the incoming and outgoing message formats matches your own.
- Second, teach your system how to understand the different message format in order to receive and send messages.
- Finally, each message should be checked as it is coming into your application during run-time in order to verify that it provides your application with the fields you require. If it does not, make sure it is put aside to be evaluated for possible noncompliance.

## Determining Message Format Differences

Determining format differences starts with examining the healthcare vendor's message specifications and sample messages. The sample of messages should be large enough to represent all types of messages you may be receiving. Using Corepoint Integration Engine, you can compare the specifications from the healthcare vendor's messages with your message format to uncover any differences.

The following figure illustrates how Corepoint Integration Engine compares a vendor's specification for a result message, to the HL7 2.2 standard definition of an ORU message. The differences between the message formats are outlined in red.

The figure is a composite image comparing a vendor's message specification with the HL7 2.2 standard definition of an ORU message. On the left, under the heading "Vendor Specification", is a snippet of HL7 message structure: MSH, PID, {[NTE]}, { ORC, OBR, { OBX, {[NTE]}, {[ZPS]}. The ZPS segment is highlighted with a red box. On the right, under the heading "Standard 2.2 Result Message Shown in Neointegrate", is a screenshot of the Neointegrate software interface. The interface shows a "Messages" tab with a "Segment Layout" section. The layout lists segments: MSH, ResultsGroup, PatientGroup, PID, {[NTE]}, [PV1], PatientGroup, OrderGroup, [ORC], OBR, {[NTE-1]}, ObservationGroup, [OBX], {[NTE-2]}, ObservationGroup, OrderGroup, ResultsGroup, and [DSC]. The segments [PV1], {[NTE-1]}, and [DSC] are highlighted with red boxes, indicating they are present in the standard but not in the vendor's specification.

As shown in the previous figure, the standard contains the following segments that the vendor's specification does not: PV1, NTE-1, and DSC. The vendor's specification contains a ZPS segment after the order detail group.

## Teaching Your Interface to Understand the Message Differences

Using the specification and sample messages, you can use Corepoint Integration Engine to create a derivative that is representative of the vendor's message format. This allows Corepoint Integration Engine to receive from and send messages in the application's desired format.

The following figure shows the same specification and a representative derivative created in Corepoint Integration Engine.

The image displays a software interface with two main panels. The left panel, titled "Vendor Specification", contains the following text:

```
1.1.1 Result Message Structure
The example below illustrates the structure of a Result message.

MSH
PID
{[NTE]}
  { ORC
    OBR
    { OBX
      {[NTE]}
    }
  }
{[ZPS]}
}
```

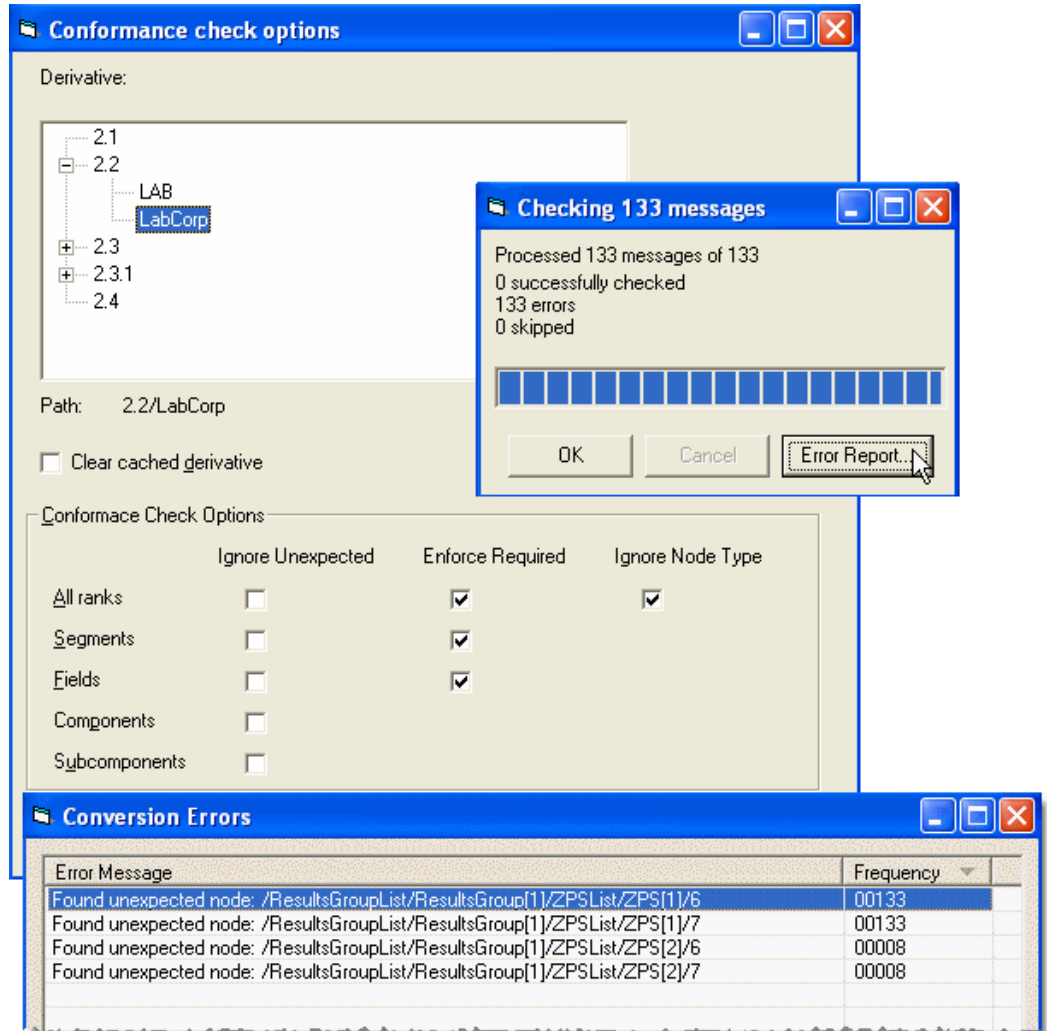
The right panel, titled "Neointegrate Derivative", is divided into three tabs: "Derivatives", "Messages", and "Segments". The "Messages" tab is active, showing a "Description" field with the text "OBSERV. RESULT/UNSO". Below this is a "Segment Layout" section with a "Segments" list containing "MSH". The "Messages" list below shows "ORU-R01" selected. At the bottom, there is a "New" button and a checkbox labeled "Copy From Selected Item".

```
MSH
( ResultsGroup
  [ PatientGroup
    PID
    {[NTE]}
  ] PatientGroup
  { OrderGroup
    [ORC]
    OBR
    { ObservationGroup
      [OBX]
      {[NTE-1]}
    } ObservationGroup
  } OrderGroup
} ResultsGroup
{[ZPS]}
```

## Letting Corepoint Integration Engine Check Message Conformance

Once you have the derivative created to match the specifications, you can use Corepoint Integration Engine to check the conformance of the sample messages against the derivative and make modifications until all the messages pass conformance.

The following figure shows the conformance checking options and error report available in Corepoint Integration Engine.

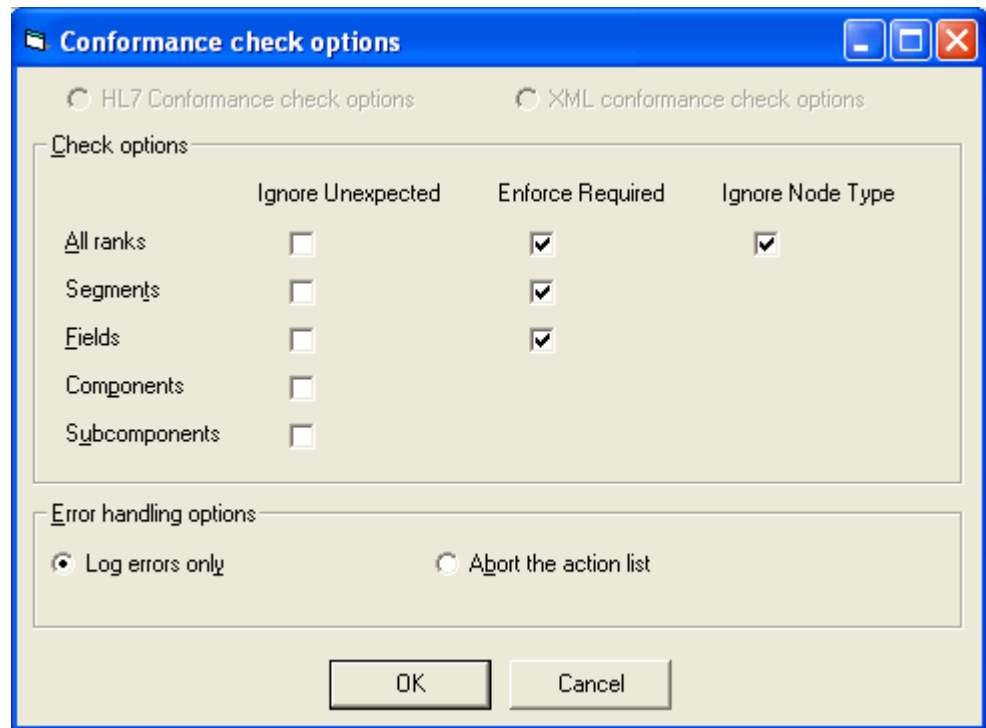


The previous figure shows a group of 133 sample messages that have been checked for conformance against a derivative created to work with the message format. The error report shows the errors with the largest number of messages affected at the top.

## Checking Messages During Run-Time

The same functionality that checks conformance in messages is also available during run-time to verify the information you need in the message is available. As shown in the following figure, there are options available that define how stringent the checks on messages coming into your system are. You must weigh both the advantages and disadvantages of how strict your conformance checking is for incoming messages. The stricter your conformance checking at run-time, the more messages will error and require manual intervention to process.

Typically you will decide to ignore the unexpected segments when you are in run-time mode, which complies with HL7's recommendation to ignore unexpected elements. You may also want to enforce required elements to prevent the receiving of incomplete data. Corepoint Integration Engine provides the flexibility to ignore unexpected message elements and enforce required message elements as shown in the following *conformance check options configuration window*.



---

## Summary

Corepoint Integration Engine can greatly reduce the time required to conformance check HL7 messages and increase accuracy by:

- Validating HL7 messages against any selected version (standard or user-modified)
- Ensuring messages comply with the desired format
- Eliminating or reducing time spent on message error correction once deployed
- Automatically checking messages against a defined custom format to find message nonconformance

## About Corepoint Health

Corepoint Health solutions deliver interoperability for healthcare organizations and simplify the complexities of healthcare data through practical software applications, consulting and training. Our innovative and proven software solutions leverage clinical data flow efficiently for a diverse group of healthcare entities including hospitals, imaging centers, laboratories, clinics and healthcare vendors. This next generation approach to healthcare data and streamlined workflow is where Corepoint Health specializes in helping customers discover the power of integration.

[www.corepointhealth.com](http://www.corepointhealth.com)

Corepoint Health  
6509 Windcrest Drive  
Suite 160B  
Plano, Texas 75024  
469-229-5000  
[info@corepointhealth.com](mailto:info@corepointhealth.com)