**Proposed Response for LOI #105: Replace current section 4.3 and 4.4 (including subsections) as follows:**

***Note that we actually do not have a profile to indicate that Acknowledgement are supported or not. So right now one must support the full end-to-end capability. In LRI these are the LRI\_Acknowledgement\_Component and the LRI\_End-to-End\_Acknowedgement\_Component. There is a an LOI\_Acknowledgement\_Component referenced in section 5.1.1, but not in section 3.2.1 where they need to be. At least LOI\_Acknowledgement Component needs to be added to 3.2.1, and we need to agree whether an LOI\_End-to-End\_Acknowledgement\_Component is necessary as well to allow for simple accept level acknowledgement as well. But that requires further changes to the below.***

4.3 Acknowledgements

This guide requires support for Acknowledgement messages to both the OML messages (whether a New and Append Order or the Cancel Order) to provide the ability to determine whether the message has been received in good order by the intended recipient. A mechanism is provided to support both node-to-node accept level acknowledgement (the receiving system has taken responsibility of the message), and the end-to-end application level acknowledgement choreography (the intended recipient not only took on responsibility of the message after the message may have passed through multiple systems such as integration engines, but can also consume the message’s application specific data). This requires the use of the Enhanced Acknowledgment Mode, i.e. MSH-15 (Accept Acknowledgment Type) and MSH-16 (Application Acknowledgement Type) are valued by the message sender and control the creation of an accept level message and an application level acknowledgement messages by the message receiver, or a node that enables transmission of the message across the various systems that may be between the sender and receiver (e.g., integration engines, HIEs, etc.). For a complete definition of an Accept Level acknowledgement and an Application Level acknowledgement, see V2.5.1 (or higher) Chapter 2.

The diagram in Figure 4‑1. LOI Message and Guaranteed Delivery Notification Flow summarizes the flow of Acknowledgements from the order sender (EHR-S) to the order receiver (LIS) and back through the different gateways.

The numbers for O = Order indicate the step in the respective flow. For example the step marked O2 indicates that for the flow of the Order message – the green arrow labeled OML and its related Accept Acknowledgement (ACK), the dotted black arrow between Gateway 2 and Gateway 1 – would be step 2.



Figure 4‑1. LOI Message and Guaranteed Delivery Notification Flow

4.3.1 Acknowledgement Choreography Applied

The acknowledgement choreography starts with the initial New and Append Order, or the Cancel Order (both using the OML^O21^OML\_O21 message) indicating in MSH-15 and MSH-16 how the receiving system is to respond. The following MSH-15 and MSH-16 values are required or permitted:

| Table 4‑5. OML acknowledgement codes  |
| --- |
| Requirement | MSH-15 | MSH-16 |
| SHALL support | AL | AL |
| MAY support | AL | ER |
| MAY support\* | NE | AL |

\*ONLY in point-to-point environments, where the transport protocol guarantees delivery to the intended recipient.

All other values and combinations are NOT allowed.

4.3.1.1 ACK^O21^ACK: Laboratory Order Message – Accept Acknowledgement

Based on the actual values in the OML^O21^OML\_O21 MSH-15 and MSH-16 values, the receiver will send an Accept Level Acknowledgement message using the following message syntax and must use the appropriate response profiles or component in MSH-21:

| Table 4‑3. ACK^O21^ACK Abstract Message Syntax |
| --- |
| Segment  | Name | Usage | Cardinality | Description |
| MSH | Message Header | R | [1..1] | The message header (MSH) segment contains information describing how to parse and process the message. This includes identification of message delimiters, sender, receiver, message type, timestamp, etc. |
|  [{SFT}] | Software Segment | O |  |  |
|  MSA | Message Acknowledgment | R | [1..1] | The Message Acknowledgment Segment (MSA) contains the information sent as an acknowledgment to the order message received by a LIS or EHR-S. |
|  [{ERR }] | Error | C(R/O) | [0..\*] | Condition Predicate: If MSA-1 (Message Acknowledgement) is not valued ‘AA’ or ‘CA’. |

This message is only used between nodes that the messages travels along per Figure 4-1. The message uses values MSA-1 Acknowledgement code to either “CA” or “CR” to the immediately preceding sender. This applies to intermediaries between a Laboratory Result Sender and an EHR-S such as HIEs and interface engines, as well as to the final EHR-S destination.

To avoid this acknowledgement from generating a response back to the originating node of the Accept Level Acknowledgement message and effectively start a never-ending series or accept acknowledgement messages between two nodes, the originating node must use the Accept Acknowledgement message (ACK^O21^ACK) with the following code combinations:

| Table 4‑6. Accept Acknowledgement Codes |
| --- |
| Requirement | MSH-15 | MSH-16 |
| SHALL support | NE | NE |

All other values and combinations are NOT allowed.

4.3.1.2 ORL^O22^ORL\_O22: Laboratory Order Message – Application Level Acknowledgement

Based on the actual values in the OML^O21^OML\_O21 MSH-15 and MSH-16 values, the receiver will send an OML^O22^ORL\_O22 Application Level Acknowledgement message using the following message syntax and must use the appropriate response profiles or component in MSH-21:

| Table 4‑8. ORL^O22^ORL\_O22 Abstract Message Syntax |
| --- |
| Segment  | Name | Usage | Cardinality | Description |
| MSH | Message Header | R | [1..1] | The message header (MSH) segment contains information describing how to parse and process the message. This includes identification of message delimiters, sender, receiver, message type, timestamp, etc. |
|  MSA | Message Acknowledgment | R | [1..1] | The Message Acknowledgment Segment (MSA) contains the information sent an acknowledgment to the order message received by a LIS or EHR-S. |
| [{ ERR }] | Error | C(R/O) | [0..\*] | Condition Predicate: If any ORC-1 (Order Control) is valued 'UC' or 'UA'. |
| [{ SFT }] | Software | O |  |  |
| [{ NTE }] | Notes and Comments (for Header) | O |  |  |
| [  | ***RESPONSE Begin*** | R | [1..1] |  |
|  [ | ***PATIENT Begin*** | R | [1..1] |  |
|  PID | Patient Identification | R | [1..1] |  |
|  [{  | ***ORDER Begin*** | R | [1..\*] |  |
|  ORC | Common Order | R | [1..1] |  |
|  [{  | ***TIMING Begin*** | O |  |  |
|  TQ1 | Timing/Quantity | R | [1..1] |  |
|  [{ TQ2 }] | Timing/Quantity Order Sequence | O |  |  |
|  }] | ***TIMING End*** |  |  |  |
|  [  | ***OBSERVATION\_REQUEST begin*** | R | [1..1] |  |
|  OBR | Observation Request | R | [1..1] |  |
|  [{  | ***SPECIMEN Begin*** | O |  |  |
|  SPM | Specimen | R | [1..1] |  |
|  [{ SAC }] | Specimen Container Details | O |  |  |
|  }] | ***SPECIMEN End*** |  |  |  |
|  ] | ***OBSERVATION\_REQUEST End*** |  |  |  |
|  }] | ***ORDER End*** |  |  |  |
|  ] | ***PATIENT End*** |  |  |  |
| ] | ***RESPONSE End*** |  |  |  |

This message provides the end-to-end delivery confirmation, including whether the receiver could consume the application specific content. It therefore is sent across all the nodes that may have been between the sender and receiver back to the originator of the New and Append Order, or the Cancel Order.

The following MSH-15 and MSH-16 values are required or permitted:

| Table 4‑7. Application Acknowledgment Codes |
| --- |
| Requirement | MSH-15 | MSH-16 |
| SHALL support | AL | NE |
| MAY support | NE | NE |

All other values and combinations are NOT allowed.

4.3.1.2.1 ACK^O22^ACK: Laboratory Order Message – Accept Acknowledgement

Based on the actual values in the OML^O22^OML\_O22 MSH-15 and MSH-16 values, the receiver will send an Accept Level Acknowledgement message using the following message syntax and must use the appropriate response profiles or component in MSH-21:

| Table 4‑3. ACK^O22^ACK Abstract Message Syntax |
| --- |
| Segment  | Name | Usage | Cardinality | Description |
| MSH | Message Header | R | [1..1] | The message header (MSH) segment contains information describing how to parse and process the message. This includes identification of message delimiters, sender, receiver, message type, timestamp, etc. |
|  [{SFT}] | Software Segment | O |  |  |
|  MSA | Message Acknowledgment | R | [1..1] | The Message Acknowledgment Segment (MSA) contains the information sent as an acknowledgment to the order message received by a LIS or EHR-S. |
|  [{ERR }] | Error | C(R/O) | [0..\*] | Condition Predicate: If MSA-1 (Message Acknowledgement) is not valued ‘AA’ or ‘CA’. |

This message is only used between nodes that the messages travels along per Figure 4-1. The message uses values MSA-1 Acknowledgement code to either “CA” or “CR” to the immediately preceding sender. This applies to intermediaries between a final EHR-S destination and an EHR-S such as HIEs and interface engines, as well as to the Laboratory Result Sender.

To avoid this acknowledgement from generating a response back to the originating node of the Accept Level Acknowledgement message and effectively start a never-ending series or accept acknowledgement messages between two nodes, the originating node must use the Accept Acknowledgement message (ACK^O21^ACK) with the following code combinations:

| Table 4‑6. Accept Acknowledgement Codes |
| --- |
| Requirement | MSH-15 | MSH-16 |
| SHALL support | NE | NE |

All other values and combinations are NOT allowed.

3/16/2017 Motion to include the changes in the LOI IG – Riki Merrick, John Roberts

Against: 0 Abstain: 0 in Favor: 8

**Proposed response for LOI #155.**

Include in the list under the diagram (page 42, just in front of 3.2.1) the full list of response components and profiles. Riki Merrick, Carolyn Knapick

Against: 0 Abstain: 0 In Favor: 5

**Proposed Response for LRI #44, #174, #175, #263, #366, #428: Replace current section 4.3 and 4.4 (including subsections) in LRI with below.**

7.3 Acknowledgements

This guide requires support for Acknowledgement messages to the ORU message to provide the ability to determine whether the message has been received in good order by the intended recipient. A mechanism is provided to support both node-to-node accept level acknowledgement (the receiving system has taken responsibility of the message), and the end-to-end application level acknowledgement choreography (the intended recipient not only took on responsibility of the message after the message may have passed through multiple systems such as integration engines, but can also consume the message’s application specific data). This requires the use of the Enhanced Acknowledgment Mode, i.e. MSH-15 (Accept Acknowledgment Type) and MSH-16 (Application Acknowledgement Type) are valued by the message sender and control the creation of an accept level message and an application level acknowledgement messages by the message receiver, or a node that enables transmission of the message across the various systems that may be between the sender and receiver (e.g., integration engines, HIEs, etc.). For a complete definition of an Accept Level acknowledgement and an Application Level acknowledgement, see V2.5.1 (or higher) Chapter 2.

Figure 7‑1. LRI Message and Guaranteed Delivery Notification Flow summarizes the flow of Acknowledgements from the results sender (LIS) to the results receiver (EHR-S) and back through the different gateways.

The numbers for R = Result indicate the step in the respective flow. For example the step marked R2 indicates that for the flow of the Result message – the solid green arrow labeled ORU and its related Accept ACK, the dotted black arrow between Gateway 2 and Gateway 1 – would be step 2.



Figure 7‑1. LRI Message and Guaranteed Delivery Notification Flow

7.3.1 Acknowledgement Choreography Applied

The acknowledgement choreography starts with the initial Results message ORU^R01^ORU\_R01 indicating in MSH-15 and MSH-16 how the receiving system is to respond. The following MSH-15 and MSH-16 values are required or permitted:

When using the basic acknowledgement profile (LRI\_Acknowledgement\_Component – ID: 2.16.840.1.113883.9.26), then the initial results message ORU^R01^ORU\_R01 SHALL support MSH-15 and MSH-16 as follows:

| Table 7-4. ORU acknowledgement codes  |
| --- |
| Requirement | MSH-15 | MSH-16 |
| SHALL support | AL | NE |
| MAY support | AL | AL |
| MAY support | AL | ER |
| MAY support\* | NE | AL |
| MAY support\* | NE | NE |
| MAY support\* | NE | ER |

\*ONLY in point-to-point environments, where the transport protocol guarantees delivery to the intended recipient.

When using the end-to-end acknowledgement profile (LRI\_END-TO-END\_Acknowledgement\_Component – ID: 2.16.840.1.113883.9.195.3.7), then the initial results message ORU^R01 SHALL support the following code combinations:

| Table 7-5. ORU acknowledgement codes  |
| --- |
| Requirement | MSH-15 | MSH-16 |
| SHALL support | AL | AL |
| MAY support | AL | ER |
| MAY support\* | NE | AL |
| MAY support\* | NE | ER |

\*ONLY in point-to-point environments, where the transport protocol guarantees delivery to the intended recipient.

All other values and combinations are NOT allowed.

7.3.1.1 ACK^R01^ACK: Results Message – Accept Acknowledgement

When MSH-21 in the initial ORU^R01^ORU\_R01 message includes either LRI\_ACKNOWLEDGEMENT\_COMPONENT (2.16.840.1.113883.9.26) or LRI\_END-TO-END\_ACNOWLEDGEMENT\_COMPONENT, and based on the actual values in the MSH-15 and MSH-16 values, the receiver shall send an Accept Level Acknowledgement message using the following message syntax and must use the appropriate response profiles or component in MSH-21, while using either “CA” or “CR in MSA-1: Acknowledgement Code. Note that due to the ACK^R01^ACK message being used for both the Accept Level and Application Level acknowledgements, the only way to distinguish whether this is an Accept Level message is the value of MSA-1.

| Table 7-3. ACK^R01^ACK Abstract Message Syntax |
| --- |
| Segment  | Name | Usage | Cardinality | Description |
| MSH | Message Header | R | [1..1] | The message header (MSH) segment contains information describing how to parse and process the message. This includes identification of message delimiters, sender, receiver, message type, timestamp, etc. |
|  [{SFT}] | Software Segment | Varies | [1..\*] | LRI\_NDBS\_Component Usage: ‘X’LRI\_PH\_Component Usage: ‘R’Usage for all other components: ‘O’ |
|  MSA | Message Acknowledgment | R | [1..1] | The Message Acknowledgment Segment (MSA) contains the information sent as acknowledgment to the result message received by an EHR-S. |
|  [{ ERR }] | Error | C(R/O) | [0..\*] | Condition predicate: If MSA-1 (Message Acknowledgement) is not valued ‘AA’ or ‘CA’.  |

Usage Notes

LRI\_PH\_Component

**SFT Segment** – The first repeat (i.e., the Laboratory Result Sender actor that generated the message) is required. Any other application that transforms the message must add an SFT segment for that application. Other applications that route or act as a conduit may add an SFT but are not required to do so. Just being "HL7 aware" is not enough to put in SFT. They actually have to manipulate the data in the transaction beyond routing.

This message is only used between nodes that the messages travels along per Figure 7-1. The message is sent only immediately preceding sender. This applies to intermediaries between a Laboratory Result Sender and an EHR-S such as HIEs and interface engines, as well as to the final EHR-S destination.

To avoid this acknowledgement from generating a response back to the originating node of the Accept Level Acknowledgement message and effectively start a never-ending series or accept acknowledgement messages between two nodes, the originating node must use the Accept Acknowledgement message (ACK^R01^ACK) with the following code combinations:

| Table 7-6. Accept Acknowledgement Codes |
| --- |
| Requirement | MSH-15 | MSH-16 |
| SHALL support | NE | NE |

All other values and combinations are NOT allowed.

7.3.1.2 ACK^R01^ACK: Results Message – Application Acknowledgement

When MSH-21 in the initial ORU^R01^ORU\_R01 message includes LRI\_END-TO-END\_ACNOWLEDGEMENT\_COMPONENT, and based on the actual values in the ORU^R01^ORU\_R01 MSH-15 and MSH-16 values, the receiver shall send an Accept Level Acknowledgement message using the following message syntax and must use the appropriate response profiles or component in MSH-21, while using either “AA”, “AE”, or “AR in MSA-1: Acknowledgement Code. Note that due to the ACK^R01^ACK message being used for both the Accept Level and Application Level acknowledgements, the only way to distinguish whether this is an Accept Level message is the value of MSA-1.

| Table 7-3. ACK^R01^ACK Abstract Message Syntax |
| --- |
| Segment  | Name | Usage | Cardinality | Description |
| MSH | Message Header | R | [1..1] | The message header (MSH) segment contains information describing how to parse and process the message. This includes identification of message delimiters, sender, receiver, message type, timestamp, etc. |
|  [{SFT}] | Software Segment | Varies | [1..\*] | LRI\_NDBS\_Component Usage: ‘X’LRI\_PH\_Component Usage: ‘R’Usage for all other components: ‘O’ |
|  MSA | Message Acknowledgment | R | [1..1] | The Message Acknowledgment Segment (MSA) contains the information sent as acknowledgment to the result message received by an EHR-S. |
|  [{ ERR }] | Error | C(R/O) | [0..\*] | Condition predicate: If MSA-1 (Message Acknowledgement) is not valued ‘AA’ or ‘CA’.  |

Usage Notes

LRI\_PH\_Component

**SFT Segment** – The first repeat (i.e., the Laboratory Result Sender actor that generated the message) is required. Any other application that transforms the message must add an SFT segment for that application. Other applications that route or act as a conduit may add an SFT but are not required to do so. Just being "HL7 aware" is not enough to put in SFT. They actually have to manipulate the data in the transaction beyond routing.

This message provides the end-to-end delivery confirmation, including whether the receiver could consume the application specific content. It therefore is sent across all the nodes that may have been between the sender and receiver back to the originator of the Results.

The following MSH-15 and MSH-16 values are required or permitted:

| Table 7-7. Application Acknowledgment Codes |
| --- |
| Requirement | MSH-15 | MSH-16 |
| SHALL support | AL | NE |
| MAY support\* | NE | NE |

\*ONLY in point-to-point environments, where the transport protocol guarantees delivery to the intended recipient.

All other values and combinations are NOT allowed.

7.3.1.2.1 ACK^O22^ACK: Laboratory Order Message – Accept Acknowledgement

Based on the actual values in the ACK^R01^ACK MSH-15 and MSH-16 values, the receiver will send an Accept Level Acknowledgement message using the following message syntax and must use the appropriate response profiles or component in MSH-21 while using either “CA” or “CR in MSA-1: Acknowledgement Code. Note that due to the ACK^R01^ACK message being used for both the Accept Level and Application Level acknowledgements, the only way to distinguish whether this is an Accept Level message is the value of MSA-1.

| Table 7-3. ACK^R01^ACK Abstract Message Syntax |
| --- |
| Segment  | Name | Usage | Cardinality | Description |
| MSH | Message Header | R | [1..1] | The message header (MSH) segment contains information describing how to parse and process the message. This includes identification of message delimiters, sender, receiver, message type, timestamp, etc. |
|  [{SFT}] | Software Segment | Varies | [1..\*] | LRI\_NDBS\_Component Usage: ‘X’LRI\_PH\_Component Usage: ‘R’Usage for all other components: ‘O’ |
|  MSA | Message Acknowledgment | R | [1..1] | The Message Acknowledgment Segment (MSA) contains the information sent as acknowledgment to the result message received by an EHR-S. |
|  [{ ERR }] | Error | C(R/O) | [0..\*] | Condition predicate: If MSA-1 (Message Acknowledgement) is not valued ‘AA’ or ‘CA’.  |

Usage Notes

LRI\_PH\_Component

**SFT Segment** – The first repeat (i.e., the Laboratory Result Sender actor that generated the message) is required. Any other application that transforms the message must add an SFT segment for that application. Other applications that route or act as a conduit may add an SFT but are not required to do so. Just being "HL7 aware" is not enough to put in SFT. They actually have to manipulate the data in the transaction beyond routing.

This message is only used between nodes that the messages travels along per Figure 7-1. The message uses values MSA-1 Acknowledgement code to either “CA” or “CR” to the immediately preceding sender. This applies to intermediaries between a final EHR-S destination and an EHR-S such as HIEs and interface engines, as well as to the Laboratory Result Sender.

To avoid this acknowledgement from generating a response back to the originating node of the Accept Level Acknowledgement message and effectively start a never-ending series or accept acknowledgement messages between two nodes, the originating node must use the Accept Acknowledgement message (ACK^O21^ACK) with the following code combinations:

| Table 7‑6. Accept Acknowledgement Codes |
| --- |
| Requirement | MSH-15 | MSH-16 |
| SHALL support | NE | NE |

All other values and combinations are NOT allowed.

3/16/2017 – Motion to include in the LRI IG, Riki Merrick, Kathy Walsh

Against: 0 Abstain: 0 In Favor: 5

**LRI #44: Proposed Disposition**

We disagree that the guide is silent on the use of Original Mode or Enhanced Mode acknowledgement (everything references Enhanced Mode: Use Case, single transactions), but agree that it could be made more clear in section 7.2 for Batch transactions. This guide uses Enhanced Mode, so the proposed wording would not work. Suggest:

* Move 7.2 after current 7.3
* Add language to state (in current 7.2):
	+ When using Acknowledgements we strongly encourage the use of enhanced mode to be consistent with individual transactions and as indicated in section 3.6.2 Batch Message, but leave this for now to the implementation trading partners to resolve.

3/16/2017 - Motion to make the moves and add the proposed language. Riki Merrick, Kathy Walsh

Against: 0 Abstain: 0 In Favor: 5