Australian Government

Department of Agriculture
and Water Resources

eResults Messaging Service
External System Interface Requirements

Approval state:	**Approved**
Version:	**1.8**
Last updated:	**20/4/2016**

Table of Contents

Document History

Amendment History

Date	Version	Description	Author
10/04/2003	1.0	Initial Published Version	Lance Christie
09/01/2004	1.1	**Section 5.2** 1) Modified the DDT_ClientJobId data type to support 9 and 10 character. 2) Removed rule that it must be filled to maximum length. **Section 5.3.2** 1) Modified the message receipt and delivery timestamp rules to support time zone and server clock disparity. **Appendix D – eResults Messaging Schema** 1) Modified the jobId attribute definition in the schema changing the minLength value from "1" to "9".	Lance Christie
31/03/2004	1.2	**Section 3.2.3** 1) Added an example of the message subject. **Appendix C – XML Examples** 1) Removed trailing milliseconds from timestamps. 2) All occurrences of the XML attribute 'responseOutcome' have been changed to outcome. 3) Removed redundant comments in examples.	Damana Madden
06/07/2004	1.3	**Appendix C – XML Examples** 1) Updated XML examples for F0001, F0002, F0005 & F0006 to reflect more realistic test scenarios and results.	Damana Madden
25/10/2004	1.4	**Section 4.4.1** 1) Added Error Code E0116 to General Business Letter Rules **Section 4.4.4 & 4.4.5** 1) Updated Message Structure for F0005 and F0006 to include optional attribute ResultRequest/@subSampleLabReportId 2) Included information about saving subSampleLabReportId and labSampleId in the System. **Section 5.1 and 5.2** Added DataEntity subSampleLabReportId and DDT_SubSampleLabReportId **Section 6.2 Amending Info** Added the word 'Fail' to the sentence – Laboratory System will still receive a Response Letter (Pass or Fail) **Section 6.3 Help Desk Support** In For Information 1) In point 5, Changed ManuallyComplete to ManuallyPass 2) Added Point 6 re Manually Fail **Appendix B**	Neelam Goel

		1) Results Business Letters Received (State Transitions) Diagram updated to show the action 'Manually Fail' **Appendix C** 1) Updated Example 5 and 7 to show how subSampleLabReportId information can be included in the XML **Appendix D** 1) Updated eResultsMessaging Schema to include optional attribute ResultRequest/@subSampleLabReportId	
17/01/2008	1.5	**Section 3.2.3** Mail Articles, Validation and Certification **Figure 1 – Mail Articles and their Specialisations** **Figure 2 - Mail Article** 1) Updated for Protective Marking policy.	Paul Young
31/08/2012	1.6	Updated logos and general text.	AIMS Admin
15/3/2013	1.7	Removed reference to F0001 and F0002 Changed AQIS references to DAFF	Imported Food
22/4/2016	1.8	Replace references to: Quarantine with biosecurity Department of Agriculture Fisheries and Forestry (DAFF) with the Department of Agriculture and Water Resources (department) Customs with Department of Immigration and Border Protection (DIBP)	Imported Food

1 Introduction

1.1 Purpose

This specification describes the characteristics of the system interface between appointed laboratory's registered laboratory computer systems (herein after referred to as 'Laboratory System') and the Department of Agriculture and Water Resources (the department) System for exchanging sample and result information.

1.2 Scope

The scope of this document is to describe the system interface requirements to support the design and implementation of the the department eResults messaging service (herein after referred to as 'eResults') between the department and Appointed Laboratories. Both the department and Laboratories registered to eResults are required to develop complementary electronic interface processes that implement the requirements as set out in this document. This will in turn become a part of Laboratory appointments.

The eResults project consists of two phases. Phase 1 will implement the eRegistration and eResults messages, and Phase 2, if feasible, will implement the eOrder messages. The message formats for Phase 2 are included for completeness but will not be implemented at this time.

Code Maintenance and Amendment messages are not in scope and will be implemented via manual processes due to the comparatively small volume and the complexity of the solution. The system will however support additional information being submitted in new messages to augment previously supplied information.

The specification defines the message concepts, process, structure and rules to implement eResults. Operational considerations are documented to ensure that the appropriate processes are implemented to support eResults.

The scope of the specification is limited to the application level interface between the systems; external System characteristics such as operating system, hardware platform and user functionality are not prescribed and therefore not specified.

1.3 Audience

1. This document should be read by anyone intending to review or implement eResults within the department and Laboratories.
2. The reader is assumed to understand the Unified Modelling Language (UML) used in this document to describe the interface, and Extensible Markup Language (XML) used to define the message formats.

1.4 Configuration Management

Appointed laboratories will receive updates to this specification as they occur.

1.5 References

This specification should be read in conjunction with the following documents:

1. *eResults External Domain Dictionary.*
2. *eResults External Code Requirements.*
3. *eResults Cyclic Redundancy Check.*

2 Imported Food Overview

2.1 Regulatory Framework

All imported food is first subject to the *Biosecurity Act 2015* and then the *Imported Food Control Act* 1992 (from here on in referred to as the 'Act'); the intent of the Act is to ensure that all imported food meets applicable standards and ensure that the food is safe for human consumption.

The department works in conjunction with Food Standards Australia New Zealand (FSANZ) who is responsible for providing advice to the department on food safety issues and administering the Australia New Zealand Food Standards Code (FSC).

Stakeholders are:

- Food Importers
- Brokers
- Department of Immigrantion and Border Protection (DIBP)
- Department of Agriculture and Water Resources
- Appointed Laboratories
- Food Standards Australia New Zealand (FSANZ)
- State and Territory Health Authorities

2.2 Imported Food Process

The following steps describe the basic process that is followed to process imported food entries that are referred to the department.

1. The Importing Party lodges an Imported Food Entry with DIBP.
2. DIBP refer an Imported Food Entry to the department based on risk profiles.
3. An initial assessment is made based on the risk and past performance and a direction is made. This includes, but is not limited to Automatic Release, Test and Hold or Release after Inspection.
4. A food control certificate is issued to the Importing Party.
5. If an inspection is required:
    1. An inspection is booked.
    2. Food is inspected and samples collected.
    3. Samples are delivered to Appointed Labs.
    4. Laboratories notify the department of Sample registration.
    5. Food is analysed by Labs and results reported.
    6. Laboratory results of analyses are assessed by the department and passed or failed. The Importer is issued with an inspection advice that gives or denies them permission to deal.

3 eResults Messaging Overview

This section introduces the eResults Messaging Service.

3.1 Messaging Framework

The eResults Messaging Service serves as the backbone to deliver the efficiencies required in processing imported food entries.

The message exchanges between the Laboratory and the deprartment will provide information and a process to ensure samples and results are tracked and assessed effectively and efficiently.

This specification is primarily concerned with the message exchange between the department and Laboratories. A description of each message and its business purpose is provided in the next section following the diagram.

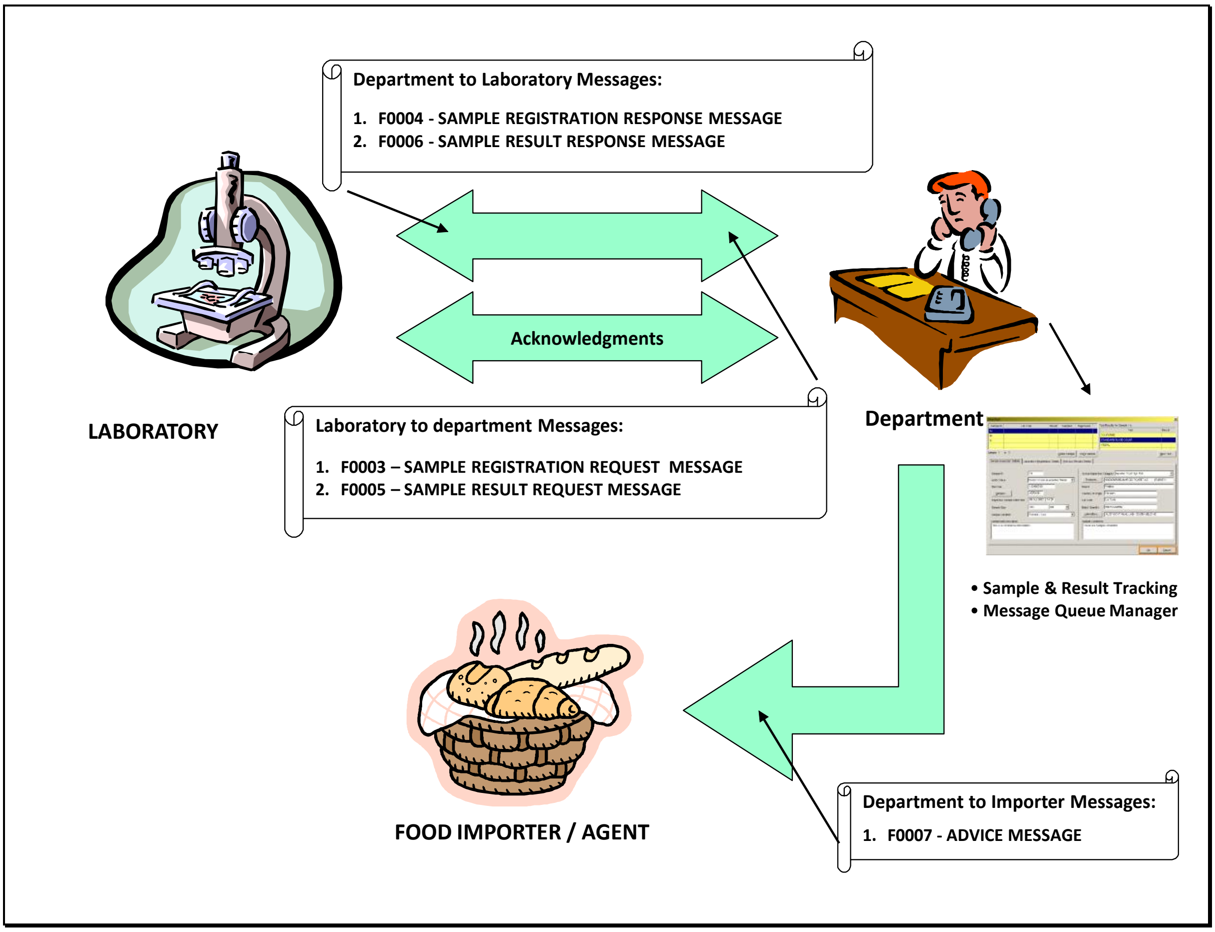

Department to Laboratory Messages:
1. F0004 - SAMPLE REGISTRATION RESPONSE MESSAGE
2. F0006 - SAMPLE RESULT RESPONSE MESSAGE
Acknowledgments
LABORATORY
Laboratory to department Messages:
1. F0003 – SAMPLE REGISTRATION REQUEST MESSAGE
2. F0005 – SAMPLE RESULT REQUEST MESSAGE
Department
• Sample & Result Tracking
• Message Queue Manager
FOOD IMPORTER / AGENT
Department to Importer Messages:
1. F0007 - ADVICE MESSAGE

Figure 3 - eResults Messaging Framework

3.1.1 Phase 1 Message Summary

Message	Benefit
eRegistration 1. **F0003** - Laboratories notify the department electronically when samples have been registered in their Laboratory System to assist in tracking and responding to importer inquiries regarding their food consignment status. 2. **F0004** - The department confirm receipt of registration information with a verification message that identifies successful and unsuccessful sample registrations.	• The department can track the location of samples to assist with Importer queries. • The department can determine if samples are delivered to Laboratories in the required timeframe and be advised of their condition. • The department gain access to timely and accurate registration information delivered with no re-keying. • Provides proof of delivery and receipt of information exchanged.
eResults 3. **F0005** - Laboratories lodge electronic results to the department at the completion of testing for the department's assessment. 4. **F0006** - The department confirm receipt of results with a verification message that identifies successful and unsuccessful sample results.	• Timely and accurate results information delivered with no re-keying. • The department can assess laboratory results and finalise imported food processing more quickly. • Results of analysis will be available department wide allowing it to be processed by any department office. • Provides proof of delivery and receipt of information exchanged.
eAdvice (Importers or their Agent) 5. **F0007** - The department publish an electronic inspection Advice to Importing Parties once the food assessment is complete that includes a summary of related laboratory results. This advice is printable.	• Provides importers with the outcome of the the department processing at various points of the imported food process.

3.1.2 eResults and the Imported Food Process

The following diagram illustrates the key activities undertaken to process an imported food entry and positions the eResults messages in context of the workflow to demonstrate how they support the activities.

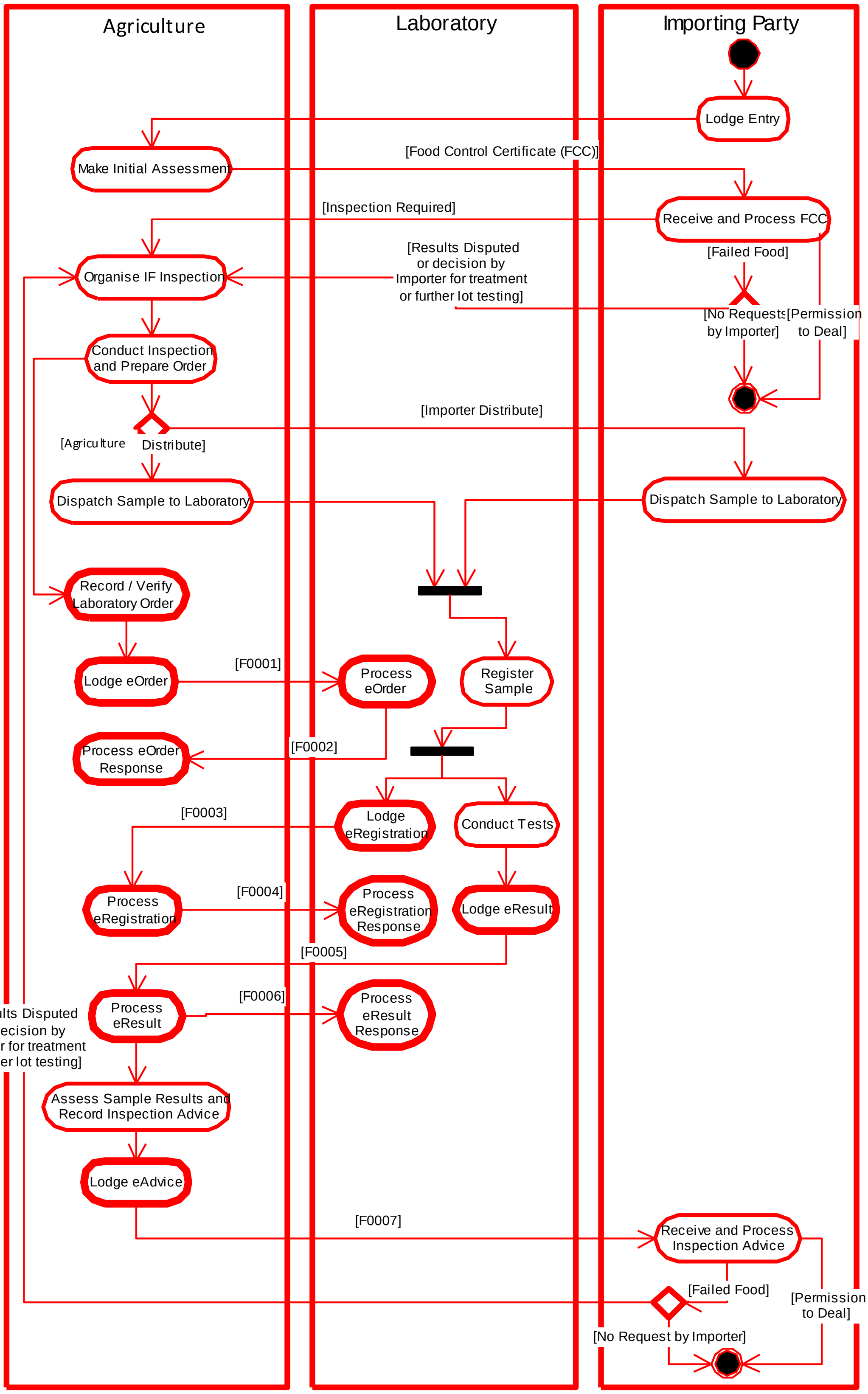


Figure 4 – eResults and the Imported Food Clearance Process

3.2 Messaging Concepts

This section will introduce the key eResults messaging concepts that support the detailed specifications.

3.2.1 Message Delivery

The physical transport and routing of eResults messages will be via unsecured Internet email using Internet Service Providers or alternatives at the discretion of the laboratory. The selected Carrier however must be capable of interfacing with the department. The method of Laboratory System connection to the carrier is a matter for resolution between the supplier of the Laboratory System and the Carrier (and is independent of the method of AIMS connection to the Carrier). Both are independent of this specification, except where Carrier message addressing requirements affect the message interchange structure.

3.2.2 eResults Enrolment and Configuration

Laboratory Systems must be enrolled with eResults as specified in their Laboratory appointment with the department. This will permit the Laboratories to implement their component of eResults and exchange eResults messages with the department. Multiple environments will facilitate development, testing and production systems.

A Laboratory Appointment will include the identification of the department eResults and Laboratory eResults email address. Laboratories will address eResults emails to the appropriate department eResults email address. The department will address eResults emails to the appropriate Laboratory eResults email address. Specific email addresses will be identified to support development, testing and production systems.

3.2.3 Mail Articles, Validation and Certification

The *Certified Mail* metaphor has been adopted by eResults as this encapsulates the delivery and acknowledgment of Messages. Envelopes are delivered, and if the addressing information is correct, they are acknowledged successfully and opened to view the message content.

In eResults, a *Mail Article* is an *Envelope* or *Acknowledgment* and is used to deliver and acknowledge business information. A Message and Mail Article are synonymous and can be used interchangeably.

A Mail Article contains a standard:

- *Subject*, which identifies the mail article type, sender id, recipient id, message id and checksum. It is used to identify key message information and enable acknowledgments.
  - *Protective Marking*, the Protective Marking Policy requires that emails be tagged with an appropriate security classification appended to the S*ubject* line. Individual classifications are subject to change on review, but all classifications are in the format [SEC=<classification>]
  - Example: Envelope,BERNHTH,eResults,244, [SEC=IN-CONFIDENCE:COMMERCIAL]
- *Address Label*, which identifies the createdTimestamp, sender id, recipient id, message id, message type id and sender and message type version. It is used to identify other key message information and enable processing.

All Mail Article exchanges between the department and Laboratory Systems must be validated before being processed to ensure that they are addressed properly, well formed and comply with standard rules. This ensures that the Mail Article is delivered to, and processed by the correct recipient.

The following diagram illustrates the eResults Mail Articles and its specialisations.

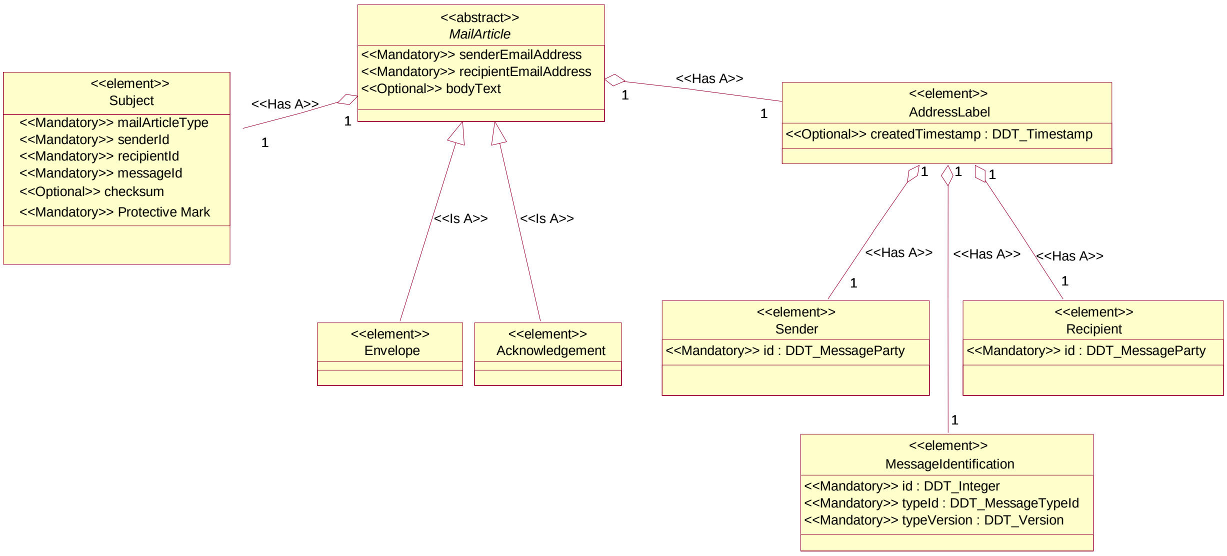

<<abstract>>
MailArticle
<<Mandatory>> senderEmailAddress
<<Mandatory>> recipientEmailAddress
<<Optional>> bodyText
<<element>>
Subject
<<Mandatory>> mailArticleType
<<Mandatory>> senderId
<<Mandatory>> recipientId
<<Mandatory>> messageId
<<Optional>> checksum
<<Mandatory>> Protective Mark
<<Has A>>
1
1
<<Has A>>
1
1
<<element>>
AddressLabel
<<Optional>> createdTimestamp : DDT_Timestamp
<<Is A>>
<<Is A>>
<<element>>
Envelope
<<element>>
Acknowledgement
<<Has A>>
<<Has A>>
<<Has A>>
1
1
1
1
1
1
<<element>>
Sender
<<Mandatory>> id : DDT_MessageParty
<<element>>
Recipient
<<Mandatory>> id : DDT_MessageParty
<<element>>
MessageIdentification
<<Mandatory>> id : DDT_Integer
<<Mandatory>> typeId : DDT_MessageTypeId
<<Mandatory>> typeVersion : DDT_Version

Figure 5 – Mail Articles and their Specialisations

3.3 Key Message Structures

There are three key Message Structures:

- *Envelopes*
- *Acknowledgments*
- *Business Letters (which are delivered in Envelopes)*

3.3.1 Envelope

Envelopes are a type of *Mail Article* and are synonymous with a letter. They are used to request (ie. initiate) or respond to a single business transaction. An *Envelope* contains a single Business Letter (defined later in this section).

Envelopes require certification before a Business Letter may be processed. This involves delivering an Acknowledgment (defined later in this section) to the sender of an Envelope. Only Envelopes that are successfully Acknowledged (PASS) are processed and the business content applied to the System.

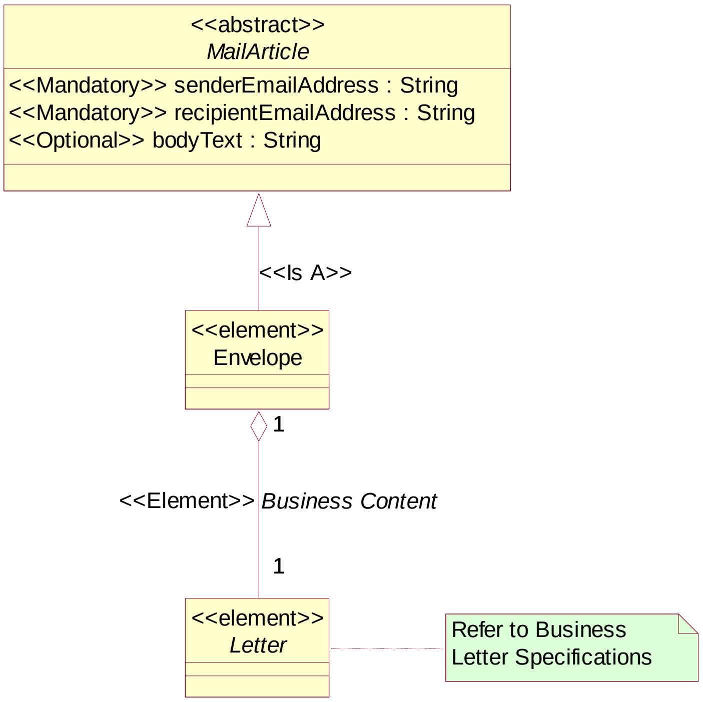

<<abstract>>
MailArticle
<<Mandatory>> senderEmailAddress : String
<<Mandatory>> recipientEmailAddress : String
<<Optional>> bodyText : String
<<Is A>>
<<element>>
Envelope
1
<<Element>> Business Content
1
<<element>>
Letter
Refer to Business Letter Specifications

Figure 6 - Envelope

3.3.2 Acknowledgment

Acknowledgments are a type of *Mail Article* and are synonymous with a *signing* for an Envelope. They are used to acknowledge the receipt of an *Envelope*. An *Acknowledgement* consists of:

Element	**Purpose**
Address Label	It is a type of Mail Article and therefore it must identify the Message, Sender and Recipient details as described earlier.
Response	If the Envelope is received and certified it is Acknowledged (PASS) and if the Envelope is received but not certified it is Acknowledgement (FAIL).
[Error]	Errors are provided back for Acknowledgment (FAIL) to enable error resolution activities. Each Error consists of an Error Code, Error Detail.

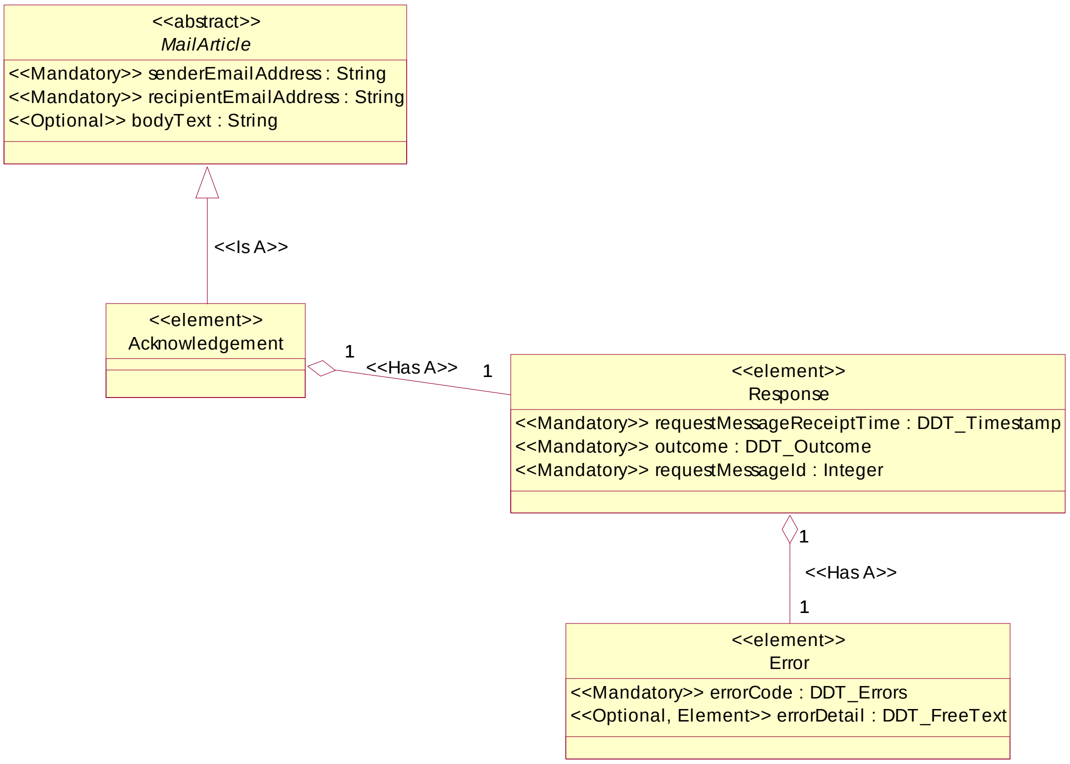

<<abstract>>
MailArticle
<<Mandatory>> senderEmailAddress : String
<<Mandatory>> recipientEmailAddress : String
<<Optional>> bodyText : String
<<Is A>>
<<element>>
Acknowledgement
1
<<Has A>>
1
<<element>>
Response
<<Mandatory>> requestMessageReceiptTime : DDT_Timestamp
<<Mandatory>> outcome : DDT_Outcome
<<Mandatory>> requestMessageId : Integer
1
<<Has A>>
1
<<element>>
Error
<<Mandatory>> errorCode : DDT_Errors
<<Optional, Element>> errorDetail : DDT_FreeText

Figure 7 - Acknowledgment

3.3.3 Business Letter

A *Business Letter* is delivered within an *Envelope*. Business Letters contain the business data that is being exchanged between the department and the Laboratory. Refer to *Section 4 Mail Article and Business Letter Specifications* for each Business Letter specification.

The following diagram illustrates that there are two types of Business Letters: a *Request Letter* and a *Response Letter*.

- Request Letters **initiate** a transaction that either requests a service (eg. order testing), or supply required information (eg. sample registrations and results).
- The corresponding Response Letter **answers** the request indicating the success or failure of the request.

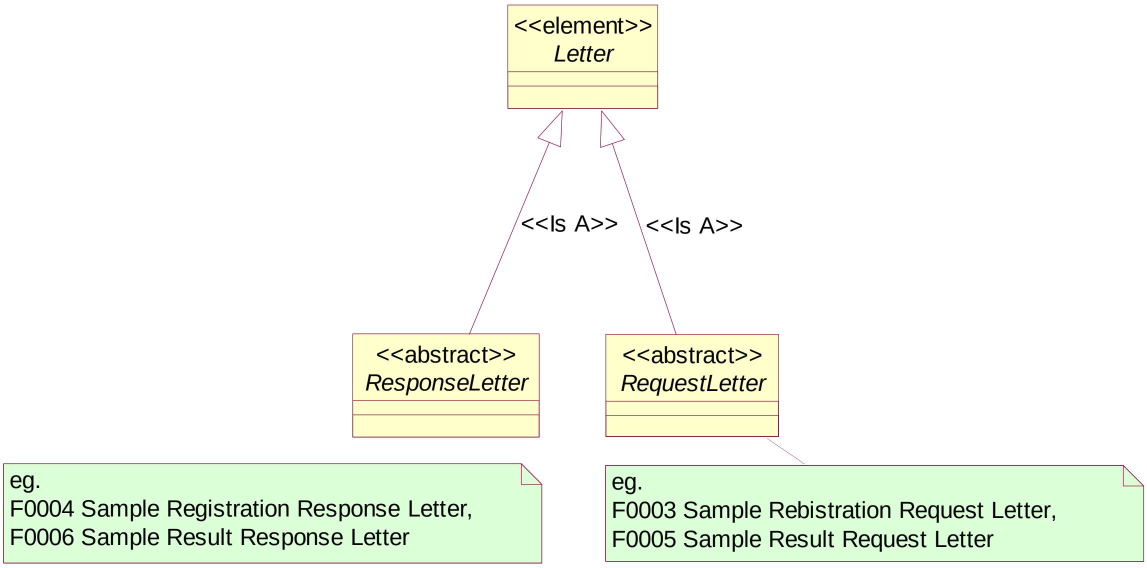

<<element>>
Letter
<<Is A>>
<<Is A>>
<<abstract>>
ResponseLetter
<<abstract>>
RequestLetter
eg.
F0004 Sample Registration Response Letter,
F0006 Sample Result Response Letter
eg.
F0003 Sample Rebistration Request Letter,
F0005 Sample Result Request Letter

Figure 8 - Business Letter

3.4 Glossary

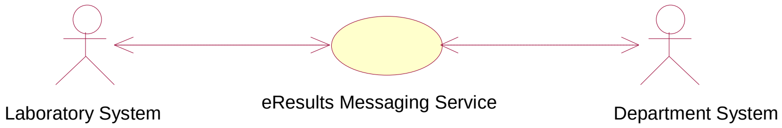
Refer to the *eResults External Domain Dictionary* for a comprehensive list of eResults terms.

3.5 Messaging Process

This section describes the key end-to-end activity of receiving, validating, acknowledging and applying eResults messages. The process will be explained using the scenario of a Laboratory System submitting eResults messages to the department.

The same process applies when the department submit eResults messages to a Laboratory System by simply reversing the roles in the use case.

3.5.1 Use Case Diagram



3.5.2 Use Case Description

This context use case details the way in which the Laboratory System (herein after referred to as 'Laboratory System') submits an eResults message (herein after referred to as 'Mail Article') to eResults (herein after referred to as the 'System'). The use case will describe the steps taken by the System and the general rules that apply to process a Mail Article. This includes the:

- Mail Article validation, certification and acknowledgement.
- Application of Business Letters to the System.
- Response Letters back to the Laboratory System.

Refer to:

- *3.2 Messaging Concepts* for background terminology and concepts.
- *4 Mail Article and Business Letter Specifications* for specifications applicable to that mail article.

3.5.3 Actors

1. A07 Laboratory System.

3.5.4 Preconditions

1. The Laboratory System is enabled and configured.
2. The department System is enabled and configured.
3. The Laboratory System is an Appointed Laboratory and is registered with the System.

3.5.5 Postconditions

1. The Mail Article is processed by the System and delivers the appropriate Acknowledgments and Response Letters.
2. The Mail Article and its contents will be saved in the System.
3. The Acknowledgments and Response Letters will be saved in the System.

3.5.6 Triggers

6. A Mail Article is received from a Laboratory System.

3.5.7 Flow of Events

1. The Laboratory System prepares and delivers one or more Mail Articles (Envelopes and Acknowledgments) to the System.
2. For each Mail Article the System:
   - 2.1 Saves the Mail Article in the System.
   - 2.2 If the Mail Article is an Envelope, the System certifies the Envelope:
     - 2.2.1 Validates the Envelope successfully.
     - 2.2.2 Certifies the Envelope by preparing and posting an Acknowledgement (PASS).
     - 2.2.3 Saves the Acknowledgement (PASS) in the System.
   - 2.3 If the Mail Article is an Acknowledgement, the System processes the Acknowledgment:
     - 2.3.1 Validates the Acknowledgement successfully.
     - 2.3.2 Marks the mail exchange as completed.
   - 2.4 If the Mail Article is an Unknown type, the System processes the Unknown:
     - 2.4.1 Rejects the Mail Article.
3. For each certified Envelope, the System (applies the Business Letter): (Refer to Section *4.4 Business Letters* for the applicable detailed specification).
   - 3.1 Saves the Business Letter in the System.
   - 3.2 If the Business Letter is a Request Letter the System:
     - 3.2.1 Confirms that the Request Letter is valid and meets all business rules.
     - 3.2.2 Applies the Request Letter to the System.

3.2.3 Prepares and Posts an Envelope containing the Response Letter (Pass).

3.2.4 Saves the Response Letter (Pass) in the System.

3.3 If the Business Letter is a Response Letter the System:

3.3.1 Confirms that the Response Letter is valid and meets all business rules.

3.3.2 Applies the Response Letter to the System and marks the business transaction as completed.

4. The System delivers the posted Mail Articles (Envelopes and Acknowledgments) to the Laboratory System.

3.5.8 Alternative Courses

Nil.

3.5.9 Exceptions

a) Envelope not Valid (Inserts After Step 2.2.1)

1. The System detects that the Envelope is not valid.
2. The System prepares and posts an Acknowledgement (FAIL) and applies the appropriate Errors. Refer to section *4.2 Envelope* for applicable rules and errors.
3. The System saves the Acknowledgement (FAIL) in the System.
4. The use case restarts at Step 2 with the next Mail Article.

b) Acknowledgment not Valid (Inserts After Step 2.3.1)

1. The System detects that the Acknowledgement is not valid.
2. The System 'Pends' the Acknowledgement and saves the Errors in the System. Refer to Section *4.3 Acknowledgment* for applicable rules and errors.
3. The use case restarts at Step 2 with the next Mail Article.

c) Request Letter not Valid - Fails General Validation (Inserts After Step 3.2.1)

1. The System detects that the Request Letter fails general validation.
2. The System prepares and posts a Response Letter (Fail) and applies the appropriate Errors. Refer to Section *4.4.1 General Business Letter Rules* for applicable rules and errors.
3. The System saves the Response Letter (Fail) in the System.
4. The use case restarts at Step 3 with the next Envelope.

d) Request Letter not Valid - Data Mismatch (Replaces 1: Steps 3.2.1 – 3.2.2 and 2: e.3.2.1)

1. The System detects that information already exists that is different to the Request Letter.
2. The System 'Pends' the Request Letter and saves the Errors in the System.
3. The use case restarts at Step 3 with the next Envelope.

e) Request Letter not Valid - Sample Does Not Exist (Replaces Steps 3.2.1 – 3.2.2)

1. The System detects that the Business Letter Sample does not exist in the operational system.
2. The System 'Holds' the Request Letter for a specified timeframe.
3. After the specified timeframe:

3.1 If the Sample cannot be matched to the System Sample, 3.1.1 The System 'Pends' the Request Letter and saves the Errors in the System.

3.1.2 The use case restarts at Step 3 with the next Envelope.

3.2 Otherwise,

3.2.1 The System Applies the Request Letter to the System.

1. The use case restarts at Step 3.2.3.

f) Response Letter Fails Validation (Replaces Steps 3.3.1)

1. The System 'Pends' the Response Letter and saves the Errors in the System. Refer to Section *4.4.1 General Business Letter Rules* for applicable rules and errors.
2. The use case restarts at Step 3 with the next Envelope.

g) Envelope not Acknowledged in Required timeframe (Inserts After Step 4)

1. The System redelivers non-acknowledged Envelopes to the Laboratory System. This will occur until the maximum retry count is reached for each Envelope.
2. The use case ends.

h) Internal Processing Error (Inserts anytime between Steps 2-4)

1. The System logs an error in the department's Monitor to be resolved manually. For example - Message not able to be saved in System.
2. The use case restarts at the next step.

3.5.10 Activity Diagram

Refer to *Appendix A Activity Diagrams*.

3.5.11 Business Rules

1. Refer to *Section 4 Mail Article and Business Letter Specifications - Mail Article Specific Rules* for rules specific to each Mail Article.
2. Refer to *Section 5 – Data Definition and Validation Rules* for general definitions and rules for each XML Entity.
3. Refer to the *eResults External Code Requirements* document Section *2.4 eError Code* for all Error descriptions.

3.5.12 Changes to Business Practices

1. Sample registrations and results of analysis will now be reported electronically to a single point.
2. All information exchanges will be acknowledged to support non-repudiation and the integrity of the imported food chain of custody.
3. Manual intervention will only be required to resolve queries or when reverting back to manual procedures due to a fault with eResults or for those Laboratories not registered to use the service.

   For Information, the department will implement:

4. *System administrators will require a message queue manager and help desk support guidelines to manage issues arising with eResults.*
5. *Imported Food Officers will use an eInbox feature to track samples and results to respond to internal and external queries. This will minimise the need to contact other internal and external people to determine the status of a consignment. The eInbox will also use Alerts to notify IF Officers of particular business exceptions that require immediate attention.*

3.5.13 Constraints

1. The eResults message exchanges must be platform independent.
2. The solution must be low cost infrastructure.
3. The solution must comply with the department’s Appointed Laboratory policy.

3.5.14 Assumptions

1. Laboratory Appointments will insist that sample registration and results of analysis will be reported to the department electronically.
2. Laboratories will continue to distribute their results of analysis directly to Food Importers (the client) as part of their commercial obligations.
3. Acknowledgements and responses indicating the outcome of processing should be provided to originator of messages to complete the exchange of electronic information.
4. It is the responsibility of the originator of a message to ensure timely delivery of messages.
5. Interim and presumptive results must be reported in the same manner as other results.
6. It is essential that all Laboratory Systems and the department have eResults code sets synchronised to avoid message failures resulting from unidentified codes. Due to the small number of trading partners and code sets, the method for synchronising codes will be achieved by the following:

- The department manually issuing emails with updates to code sets to Laboratories.
- Laboratories manually updating their system reference tables.

7. The System will not support the modification of information already received and processed. The System will only support additional information being submitted in new messages to augment previously supplied information. Deleting information is not permitted but may be *deactivated* manually in the operational system.
8. One and only one production messaging mailbox will be established for each Laboratory Appointment.
9. Certain events will prevent eResults messages to be processed automatically requiring manual intervention to complete the processing.
10. XML schemas are the best method to define and validate message structures, data types and rules.

3.5.15 Notes

1. Errors ranging from **E0001** to **E0099** are concerned with message delivery and general validation.
2. Errors ranging from **E0100** to **E9999** are concerned with specific business content validation and business rules.
3. The XML parsing and schema validation of a message will generate the general summary message - '**E0004 - The XML attachment must be well formed XML and conform to certified Message schemas**':
4. All Business Letters (Request Letters and Response Letters) will be sent in individual Envelopes and must be acknowledged individually.
5. All *Envelopes* must be *Acknowledged*.
6. All *Request Letters* must be answered with a *Response Letter*.
7. A *Response Letter* will include the associated *Request Letter* and an *Outcome*.
8. A *Response Letter (Fail)* will include an Error element.
9. An *Acknowledgment (Fail)* will include an Error element.
10. The first error detected will be reported and the processing terminated to avoid consequential error reporting.

3.5.16 Examples and Schema Definition

Refer to Appendix C for XML examples.

Refer to Appendix D for XML Schema.

4 Mail Article and Business Letter Specifications

This section provides a specification for each of the Mail Articles that can be exchanged in the eResults messaging service. Refer to the following references to assist in understanding each Mail Article specification:

1. *eResults Code Requirements* document for all Error descriptions.
2. Section *5 – Data Definition and Validation Rules* for XML Entity general rule definitions.
3. Section *Mail Article Specific Rules* in each Mail Article specification for rules specific to the Mail Article.

4.1 Mail Article

This section describes the specifications for Mail Articles.

1. Description

In eResults, a *Mail Article* is an *Envelope* or *Acknowledgment* and is used to deliver and acknowledge business information. It consists of addressing information and XML content.

2. Responsibilities

a) Mail Article Creator

1. Prepare and Post the Mail Article complying with the business rules specified in this section.

b) Mail Article Certifier

1. Validate that the Envelope complies with the business rules specified in this section.
2. Certify the Envelope by preparing and posting an Acknowledgement.

3. Message Structure

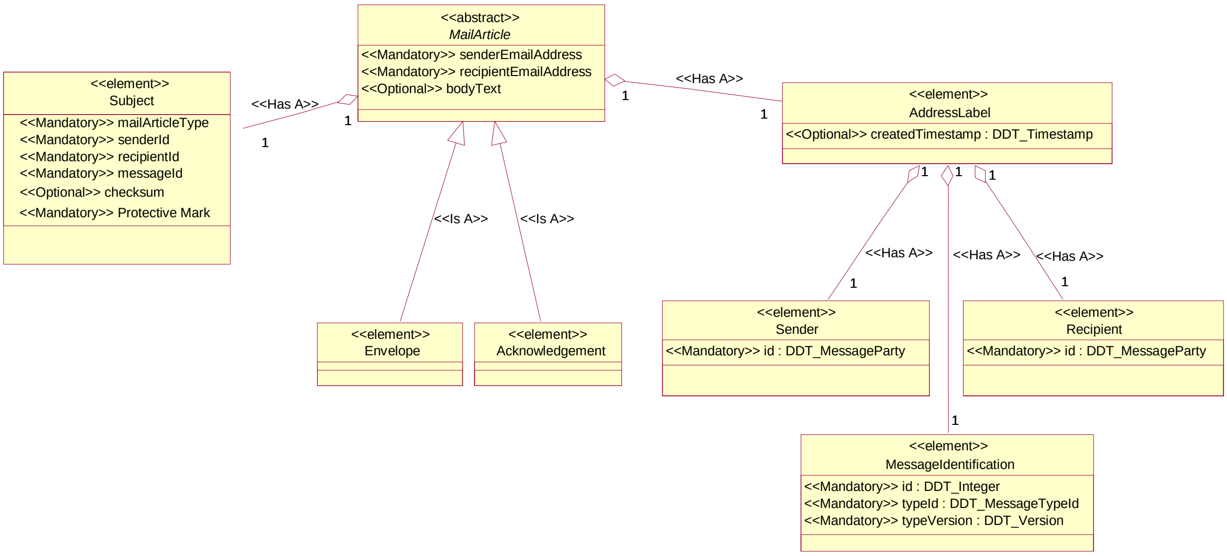

<<abstract>>
MailArticle
<<Mandatory>> senderEmailAddress
<<Mandatory>> recipientEmailAddress
<<Optional>> bodyText
<<element>>
Subject
<<Mandatory>> mailArticleType
<<Mandatory>> senderId
<<Mandatory>> recipientId
<<Mandatory>> messageId
<<Optional>> checksum
<<Mandatory>> Protective Mark
<<Has A>>
1
1
<<Has A>>
1
1
<<element>>
AddressLabel
<<Optional>> createdTimestamp : DDT_Timestamp
<<Is A>>
<<Is A>>
<<element>>
Envelope
<<element>>
Acknowledgement
1
1
1
<<Has A>>
<<Has A>>
<<Has A>>
1
1
<<element>>
Sender
<<Mandatory>> id : DDT_MessageParty
<<element>>
Recipient
<<Mandatory>> id : DDT_MessageParty
1
<<element>>
MessageIdentification
<<Mandatory>> id : DDT_Integer
<<Mandatory>> typeId : DDT_MessageTypeId
<<Mandatory>> typeVersion : DDT_Version

Figure 9 - Mail Article

4. Mail Article Specific Rules

Department processing will ignore the bodyText but will validate the subject.

Business Rule	Error Code
1. The Mail Article Subject must be valid.	**E0001**
2. The Mail Article must contain only one Attachment.	**E0002**
3. The Mail Article Subject Checksum must match the checksum calculated by the Recipient.	**E0005**
4. The Mail Article Type must be either ‘Envelope’ or ‘Acknowledgement’.	**E0013**
5. The Mail Article Subject SenderId and RecipientId must be supported by the Recipient.	**E0010**

The follow business rules are checked:

- The MailArticleType must be either "Envelope" or ""Acknowledgement".
- The SenderId and RecipientId must be supported by the receiving system.
- Checksum is verified against the data content, if provided.

4.2 Envelope

This section describes the specifications for Envelopes.

5. Description

Envelopes are a type of *Mail Article* that is synonymous to an every day letter that people mail. They are the mechanism to deliver single *Business Letter*, which requests (ie. initiates) or responds to a single business transaction.

Envelopes must be validated against the XML schema, see *Appendix D, XML Schema Definition – ‘a) Certified Mail’*.

Envelopes must then be certified, which involves preparing and posting an Acknowledgment (defined next in Section *4.3 Acknowledgment*). This process then allows the Recipient to process the Business Letter.

6. Responsibilities

a) Envelope Creator

1. Prepare and Post the Envelope complying with the business rules specified in this section.

b) Envelope Certifier

1. Validate that the Envelope complies with the business rules specified in this section.
2. Certify the Envelope by preparing and posting an Acknowledgement.

7. Message Structure

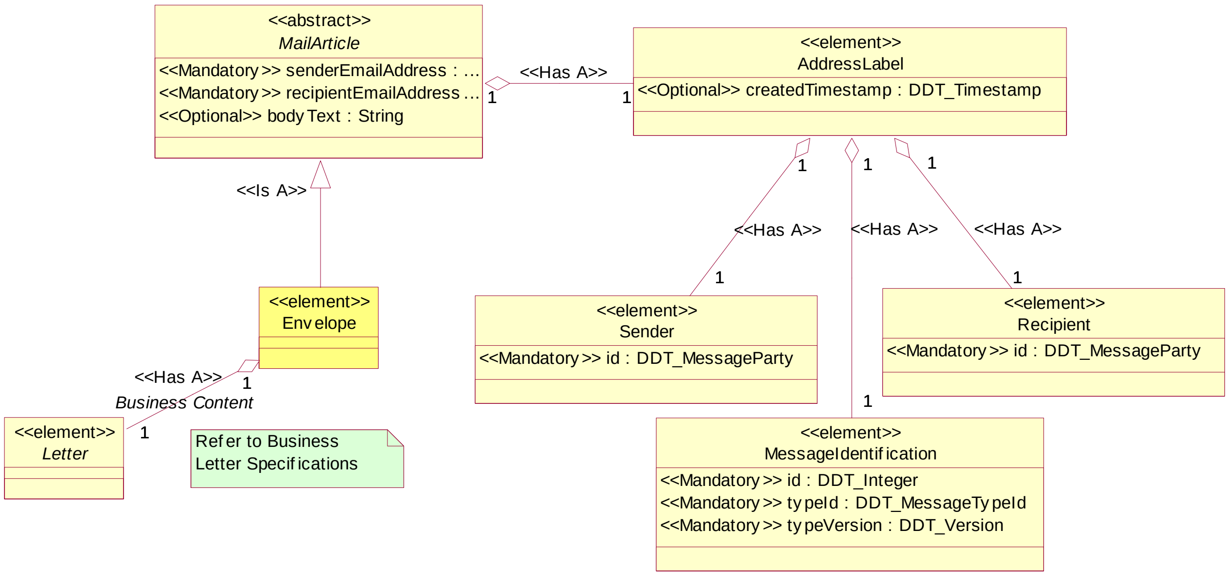

<<abstract>>
MailArticle
<<Mandatory>> senderEmailAddress : ...
<<Mandatory>> recipientEmailAddress ...
<<Optional>> bodyText : String
<<Has A>>
1
1
<<element>>
AddressLabel
<<Optional>> createdTimestamp : DDT_Timestamp
<<Is A>>
<<element>>
Envelope
<<Has A>>
1
Business Content
<<element>>
Letter
1
Refer to Business Letter Specifications
<<Has A>>
<<Has A>>
<<Has A>>
1
1
1
1
1
1
<<element>>
Sender
<<Mandatory>> id : DDT_MessageParty
<<element>>
Recipient
<<Mandatory>> id : DDT_MessageParty
<<element>>
MessageIdentification
<<Mandatory>> id : DDT_Integer
<<Mandatory>> typeId : DDT_MessageTypeId
<<Mandatory>> typeVersion : DDT_Version

Figure 10 - Envelope

8. Mail Article Specific Rules

a) Envelope Validation

The following table describes all rules to be enforced during Envelope validation.

Business Rule	Error Code
1. The XML must be well-formed.	**E0003**
2. The XML must conform to the relevant XML schema.	**E0004**
3. Created timestamp must be no later than the time of reception of the message.	**E0006**
4. The combination of Sender Id, Message Type Id and Message Type Version must be supported by the Recipient.	**E0007**
5. The Mail Article Address Label must match the Mail Article Subject.	**E0011**

b) Address Label Requirements

The following table specifies the values required when creating an Envelope for each of the Business Letters.

XML Entity	**F0003**	**F0004**	**F0005**	**F0006**
createdTimestamp	Current System timestamp generated at message creation	Current System timestamp generated at message creation	Current System timestamp generated at message creation	Current System timestamp generated at message creation
Sender/@id	Valid Laboratory Enrolment Id	‘eResults’	Valid Laboratory Enrolment Id	‘eResults’
Recipient/@id	‘eResults’	Request Letter - Sender/@id	‘eResults’	Request Letter - Sender/@id
MessageIdentification/id	Laboratory System’s next unique sequential message id for eResults	eResults (department) next unique sequential message id associated with the Laboratory System Recipient/@id	Laboratory System’s next unique sequential message id for eResults	eResults (department) next unique sequential message id associated with the Laboratory System Recipient/@id
typeId	F0003	F0004	F0005	F0006
typeVersion	Valid eResults Version - initially ‘1.0’	Valid eResults Version - initially ‘1.0’	Valid eResults Version - initially ‘1.0’	Valid eResults Version - initially ‘1.0’

4.3 Acknowledgment

This section describes the specifications for Acknowledgements.

9. Description

Every Envelope received must be acknowledged to notify the Sender that it was or was not received successfully.

If the Envelope received is valid an Acknowledgment (ACK) is delivered back to the Sender. This process finalises the Mail Exchange for that Envelope and allows the Recipient to process the Business Letter.

If the Envelope received is not valid an Acknowledgment (NACK) is delivered back to the Sender and the error(s) identified to permit error resolution activities. Each Error consists of an Error Code and Error Detail.

10. Responsibilities

a) Acknowledgment Creator

1. Create the Acknowledgement complying with the business rules specified in this section.

b) Acknowledgment Processor

1. Validate that the Acknowledgement complies with the business rules specified in this section.
2. Process the Acknowledgement to complete the mail exchange.

11. Message Structure

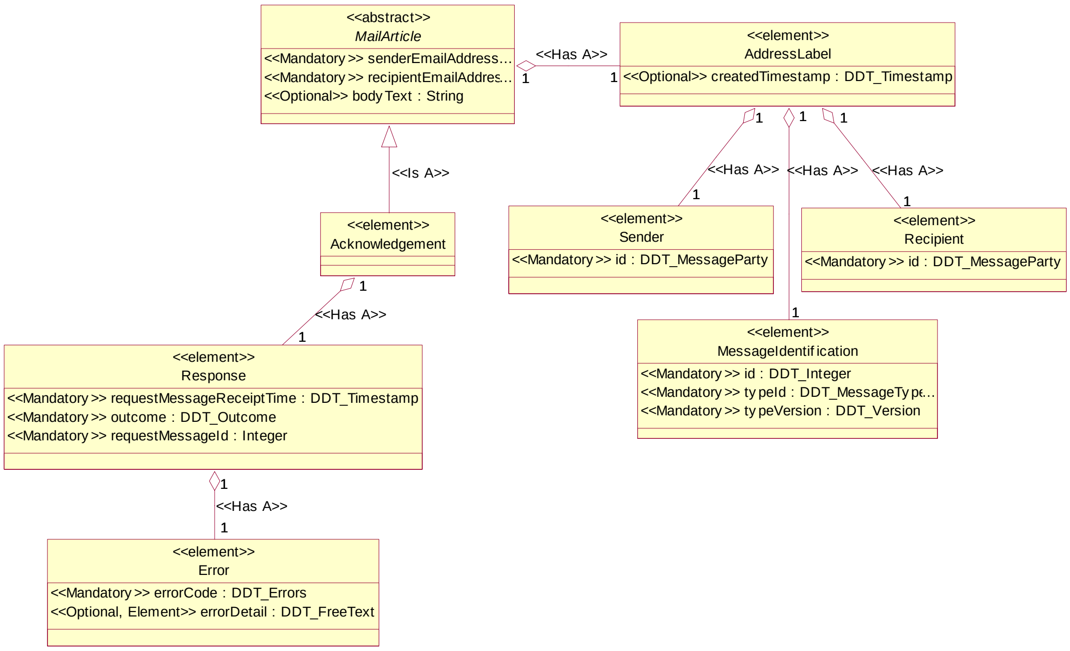

<<abstract>>
MailArticle
<<Mandatory >> senderEmailAddress...
<<Mandatory >> recipientEmailAddres...
<<Optional>> body Text : String
<<Has A>>
1
1
<<element>>
AddressLabel
<<Optional>> createdTimestamp : DDT_Timestamp
<<Is A>>
<<element>>
Acknowledgement
1
<<Has A>>
1
<<Has A>>
1
<<Has A>>
1
<<Has A>>
1
<<element>>
Sender
<<Mandatory >> id : DDT_MessageParty
<<element>>
Recipient
<<Mandatory >> id : DDT_MessageParty
1
<<element>>
MessageIdentification
<<Mandatory >> id : DDT_Integer
<<Mandatory >> typeId : DDT_MessageType...
<<Mandatory >> typeVersion : DDT_Version
1
<<Has A>>
1
<<element>>
Response
<<Mandatory >> requestMessageReceiptTime : DDT_Timestamp
<<Mandatory >> outcome : DDT_Outcome
<<Mandatory >> requestMessageId : Integer
1
<<Has A>>
1
<<element>>
Error
<<Mandatory >> errorCode : DDT_Errors
<<Optional, Element>> errorDetail : DDT_FreeText

Figure 11 - Acknowledgment

12. Mail Article Specific Rules

a) Acknowledgment Validation

The following table describes all rules to be enforced during Acknowledgement validation.

Business Rule	Error Code
1. The XML must be well-formed.	**E0003**
2. The XML must conform to the relevant XML schema.	**E0004**
3. Created timestamp must be no later than the time of reception of the message.	**E0006**
4. The Mail Article Address Label must match the Mail Article Subject.	**E0011**
5. Errors must be specified if Response Outcome is Fail.	**E0012**

b) Address Label Requirements

The following table specifies the values required when creating an Acknowledgment.

XML Entity	R0002 Department	R0002 Laboratory
createdTimestamp	Current System timestamp generated at message creation	Current System timestamp generated at message creation
Sender/@id	‘eResults’	Valid Laboratory Enrolment Id
Recipient/@id	Valid Laboratory Enrolment Id	‘eResults’
MessageIdentification/id	eResults (department) next unique sequential message id associated with the Laboratory System Recipient/@id	Laboratory System’s next unique sequential message id for eResults
typeId	R0002	R0002
typeVersion	Valid eResults Version - initially ‘1.0’	Valid eResults Version - initially ‘1.0’

4.4 Business Letters

This section describes the specifications for Business Letters and general business rules.

4.4.1 General Business Letter Rules

The following table describes all rules to be enforced during Business Letter validation. The Envelope is assumed valid at this point.

Business Rule	Error Code
1. The XML must be well-formed.	**E0003**
2. The XML must conform to the relevant XML schema.	**E0004**
3. Data dependencies require that this group of fields all be populated.	**E0009**
4. Errors must be specified if Response Outcome is Fail.	**E0012**
5. The reference code specified must be a valid reference value.	**E0100**
6. Dates must be in chronological order.	**E0101**
7. Test Results sent to the department should correspond to Tests requested by the department.	**E0102**
8. Client Job Id in message must exist in AIMS.	**E0103**
9. Client Job Id must not be finalised to receive data.	**E0104**
10. Line must not be food finalised to receive data.	**E0105**
11. Client Job Id and Client Sample Id combination in message must exist in AIMS.	**E0106**
12. Client Job Id, Client Sample Id and Test Code combination in message must exist in AIMS.	**E0107**
13. The sending Laboratory must match the Laboratory assigned to the Sample (in AIMS).	**E0108**
14. The Result Name must be the ordered Test Code for Single Result Test Types.	**E0109**
15. Response letters must contain a complete and exact copy of the corresponding Request letter.	**E0110**
16. The Business Letter Message Type Id must be the same as the Message Type Id in the Envelope Address Label.	**E0111**
17. The combination of Message Type Id and Message Type Version must be supported by the Recipient.	**E0112**
18. A Single Result Type Test must only have one result of analysis.	**E0113**
19. Each Object must be uniquely identifiable within the Object’s scope.	**E0114**
20. Manually Failed.	**E0116**

For Information:

21. *Request Letters may be ‘Held’ by the System for a specified timeframe and then ‘Pended’ if it cannot be matched to a Sample.*

22. *Request Letters may be 'Pended' by the System if it detects information that already exists in the operational system is different to the content of the Letter. The system will process 'Held' and 'Pended' Request Letters in the following way:*
    - *The department will attempt to process Held Request Letters periodically to provide time for manual processes to be completed to receive the information. The System will Pend the Request Letters after a specified timeframe.*
    - *The Message Queue Management System will identify messages that are 'Held' and 'Pended' to assist in the efficient processing of Request Letters.*
    - *The Message Queue Management System will allow applicable the department staff to submit 'Pended' messages for final processing in order to remove the 'Pended' status and generate the Response Letter.*

4.4.2 F0003 – Sample Registration Request Letter

13. Description

Analytical Laboratories that are appointed by the department are required to notify the department of samples delivered to their laboratory for testing. Laboratories must submit the Sample Registration Request Letter to the department, which identifies an imported food sample received by Laboratories for analytical testing.

14. Responsibilities

a) Laboratory

1. Register Client Samples from a Client order.
2. Generate the Sample Registration Request Letter from their Laboratory System complying with the eResults business rules and delivery timeframes as specified in their appointment.
3. Deliver the Sample Registration Request Letter to the the department's eResults messaging service.
4. Track Sample Registration Request Letters sent for acknowledgement.

b) Department

1. Receive, acknowledge and process the electronic Sample Registration Request Letter.
2. Track Sample Registration Request Letters received and processed for future reference.

15. Message Structure

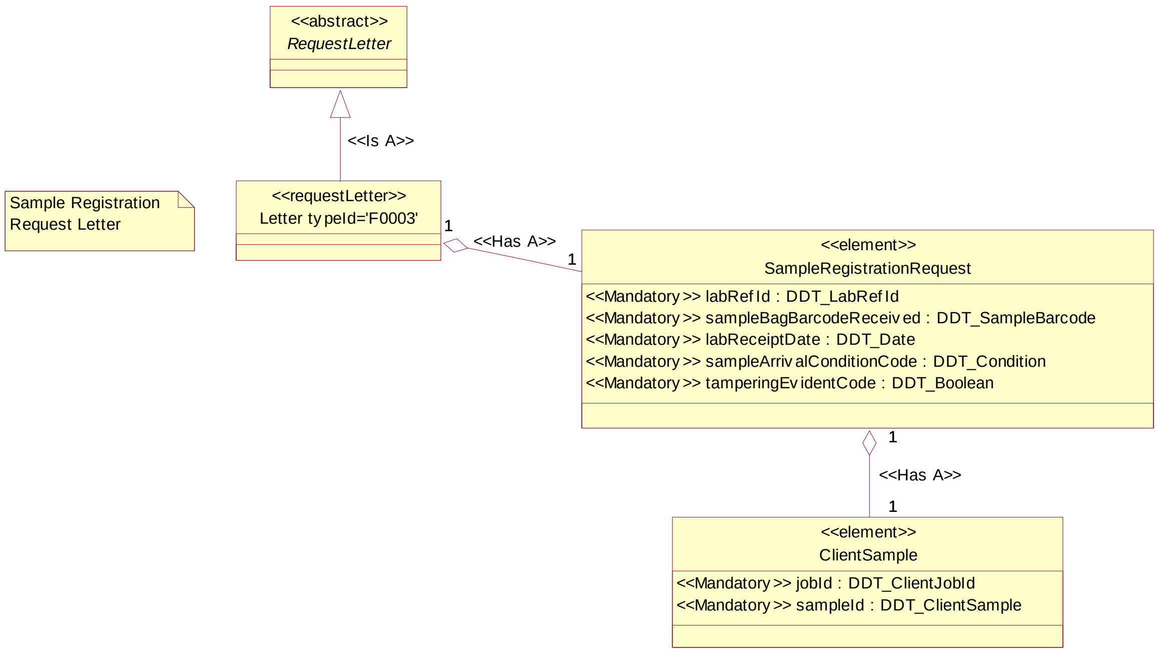

<<abstract>>
RequestLetter
<<Is A>>
Sample Registration Request Letter
<<requestLetter>>
Letter typeId='F0003'
1
<<Has A>>
1
<<element>>
SampleRegistrationRequest
<<Mandatory>> labRefId : DDT_LabRefId
<<Mandatory>> sampleBagBarcodeReceived : DDT_SampleBarcode
<<Mandatory>> labReceiptDate : DDT_Date
<<Mandatory>> sampleArrivalConditionCode : DDT_Condition
<<Mandatory>> tamperingEvidentCode : DDT_Boolean
1
<<Has A>>
1
<<element>>
ClientSample
<<Mandatory>> jobId : DDT_ClientJobId
<<Mandatory>> sampleId : DDT_ClientSample

Figure 12 - Sample Registration Request Letter Structure

16. Mail Article Specific Rules

1. A sample registration is uniquely identified by [ClientSample/@jobId + ClientSample/@sampleId].
2. One Sample Registration Request Letter corresponds to one Sample.

17. Assumptions

1. The Envelope is valid.
2. The ClientSample/@jobId is the Customs Entry Number and is returned from the Order or Order Request Letter.
3. The System will generate and record the Receipt Timestamp of the Sample Registration Request Letter at the department.
4. The Order Request Letter will be implemented in Phase 2.

4.4.3 F0004 – Sample Registration Response Letter

18. Description

The department is required to notify the Laboratory of the outcome of the Sample Registration that is contained in the corresponding Sample Registration Request Letter. The Sample Registration Response Letter is generated at the completion of the capture and processing of the Sample Registration Request Letter.

19. Responsibilities

a) Department

1. Process the Sample Registration Request Letter
   - Validate that the message complies with specified business rules.
   - Validate that the message complies with the delivery timeframes as specified in the Laboratory appointment.
   - Update department's operational system (ie. AIMS) with the validated data.
2. Generate the Sample Registration Response Letter specifying the Outcome of Request Letter.
3. Deliver the Sample Registration Response Letter to the Laboratory System messaging service.
4. Track Sample Registration Response Letters sent.

b) Laboratory

1. Receive the electronic Sample Registration Response Letter.
2. Track Sample Registration Response Letters received.
3. Process the Sample Registration Response Letter:
   - <Optional> Update Laboratory System with status information.
   - Perform remedial action on unsuccessful information rejected by the department.

20. Message Structure

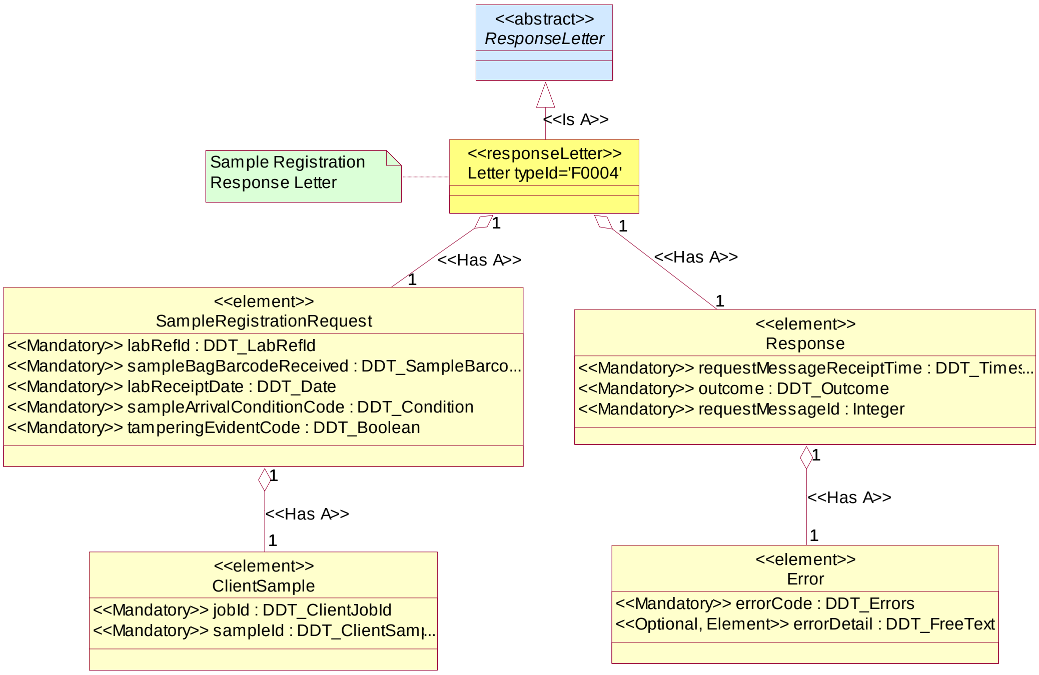

<<abstract>>
ResponseLetter
<<Is A>>
Sample Registration Response Letter
<<responseLetter>>
Letter typeId='F0004'
1
<<Has A>>
1
1
<<Has A>>
1
<<element>>
SampleRegistrationRequest
<<Mandatory>> labRefId : DDT_LabRefId
<<Mandatory>> sampleBagBarcodeReceived : DDT_SampleBarco...
<<Mandatory>> labReceiptDate : DDT_Date
<<Mandatory>> sampleArrivalConditionCode : DDT_Condition
<<Mandatory>> tamperingEvidentCode : DDT_Boolean
<<element>>
Response
<<Mandatory>> requestMessageReceiptTime : DDT_Times...
<<Mandatory>> outcome : DDT_Outcome
<<Mandatory>> requestMessageId : Integer
1
<<Has A>>
1
1
<<Has A>>
1
<<element>>
ClientSample
<<Mandatory>> jobId : DDT_ClientJobId
<<Mandatory>> sampleId : DDT_ClientSamp...
<<element>>
Error
<<Mandatory>> errorCode : DDT_Errors
<<Optional, Element>> errorDetail : DDT_FreeText

Figure 13 - Sample Registration Response Letter Structure

21. Mail Article Specific Rules

1. A sample registration is uniquely identified by [ClientSample/@jobId + ClientSample/@sampleId].
2. One Sample Registration Response Letter corresponds to one Sample.
3. An Error is uniquely identified by [Error/@errorCode].
4. The Response Letter will reference a copy of the original Request Letter - AddressLabel and SampleRegistrationRequest elements.
5. If the SampleRegistrationRequest is processed without error the Response/@outcome will be set to 'Pass' (ie. Sample Registration accepted and passed).
6. If the SampleRegistrationRequest is processed with an error the Response/@outcome will be set to 'Fail' (ie. Sample Registration failed) and the appropriate error must be supplied in the Error element.

22. Assumptions

1. The Envelope is valid.
2. The Response Letter Recipient/@id plus the Response/@requestMessageId uniquely identify the corresponding Request Message.
3. The Response/@requestMessageReceiptTime was created when processing the Request Letter.
4. Those Samples that are rejected will need to be resubmitted by the Laboratory in a subsequent message to be processed.

4.4.4 F0005 – Sample Result Request Letter

23. Description

Analytical Laboratories that are appointed by Imported Food are required to notify the department of results produced from analytical testing. Laboratories must submit the Sample Result Request Letter to the department, which identifies samples, tests and results of analysis for tests ordered.

24. Responsibilities

a) Laboratory

1. Generate the Sample Result Request Letter from their Laboratory System complying with the eResults business rules and delivery timeframes as specified in their appointment.
2. Deliver the Sample Result Request Letter to the the department's eResults messaging service.
3. Track Sample Result Request Letters sent for acknowledgement.

b) Department

1. Receive, acknowledge and process the electronic Sample Result Request Letter.
2. Track Sample Result Request Letters received and processed for future reference.

25. Message Structure

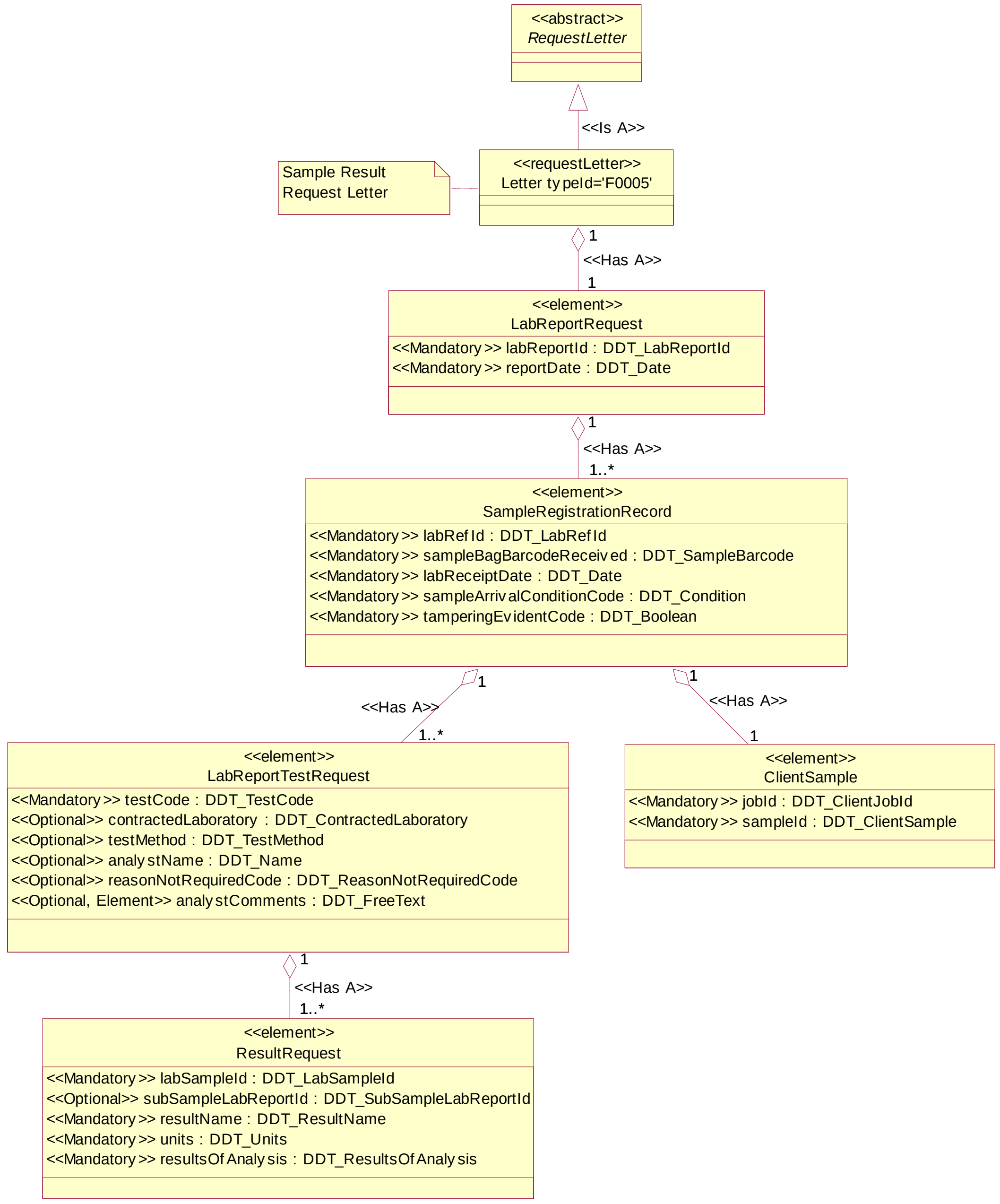

<<abstract>>
RequestLetter
<<Is A>>
Sample Result Request Letter
<<requestLetter>>
Letter typeId='F0005'
1
<<Has A>>
1
<<element>>
LabReportRequest
<<Mandatory>> labReportId : DDT_LabReportId
<<Mandatory>> reportDate : DDT_Date
1
<<Has A>>
1..*
<<element>>
SampleRegistrationRecord
<<Mandatory>> labRefId : DDT_LabRefId
<<Mandatory>> sampleBagBarcodeReceived : DDT_SampleBarcode
<<Mandatory>> labReceiptDate : DDT_Date
<<Mandatory>> sampleArrivalConditionCode : DDT_Condition
<<Mandatory>> tamperingEvidentCode : DDT_Boolean
1
<<Has A>>
1..*
1
<<Has A>>
1
<<element>>
LabReportTestRequest
<<Mandatory>> testCode : DDT_TestCode
<<Optional>> contractedLaboratory : DDT_ContractedLaboratory
<<Optional>> testMethod : DDT_TestMethod
<<Optional>> analystName : DDT_Name
<<Optional>> reasonNotRequiredCode : DDT_ReasonNotRequiredCode
<<Optional, Element>> analystComments : DDT_FreeText
<<element>>
ClientSample
<<Mandatory>> jobId : DDT_ClientJobId
<<Mandatory>> sampleId : DDT_ClientSample
1
<<Has A>>
1..*
<<element>>
ResultRequest
<<Mandatory>> labSampleId : DDT_LabSampleId
<<Optional>> subSampleLabReportId : DDT_SubSampleLabReportId
<<Mandatory>> resultName : DDT_ResultName
<<Mandatory>> units : DDT_Units
<<Mandatory>> resultsOfAnalysis : DDT_ResultsOfAnalysis

Figure 14 – Sample Result Request Letter Structure

26. Mail Article Specific Rules

1. A LabReportRequest is uniquely identified by [labReportId].
2. One Sample Result Request Letter corresponds to one Laboratory Report.
3. A SampleRegistrationRecord is uniquely identified by [ClientSample/@jobId + ClientSample/@sampleId].
4. A LabReportTestRequest is uniquely identified by [testCode].
5. A ResultRequest is uniquely identified by [labSampleId + resultName].
6. For a LabReportTestRequest/@testCode that is a 'Single Result Test', eg. *Aflatoxin*:

- The System will permit one and only one ResultRequest element. **<E0118>**
- The ResultRequest/@resultName must be LabReportTestRequest/@testCode. **<E0109 >**

7. For a LabReportTestRequest/@testCode that is a 'Multiple Result Test' such as *Pesticides*:
    - The System will allow one or more ResultRequest elements may be supplied with a unique [labSampleId + resultName]

27. Assumptions

1. The Envelope is valid.
2. The ClientSample/@jobId is the Customs Entry Number and is returned from the Order or Order Request Letter.
3. Although not enforced by the System, a LabReportTestRequest/@contractedLaboratory must be supplied if the Test ordered by the department was outsourced. They must be NATA accredited to perform the ordered test.
4. A Laboratory or Contracted Laboratory must perform all analysis for a given ordered test for a given sample - even if the test for that sample is reported more than once.
5. If ResultRequest/@units are not required as part of a Result then 'N/A' should be supplied in this attribute.
6. Laboratories are only required to provide LabReportTestRequest/@analystComments at the Test level on the report.
7. LabReportTestRequest/@analystName should be formatted: *first name + surname*.

For Information:

8. *For 'Merge-able Test Types' such as Micros the department will process as follows:*
    - *Each 'ResultRequest/@resultsOfAnalysis' will be merged comma separated into a single Results of Analysis value in AIMS.*
    - *Each 'ResultRequest/@labSampleId' will be merged comma separated into a single Lab Sample Id value in AIMS.*
    - *Each 'ResultRequest/@labSampleId' and 'ResultRequest/@subSampleLabReportId' will be saved in the System where a value for 'ResultRequest/@subSampleLabReportId' is provided.*
9. *The System will generate and record the Receipt Timestamp of the Sample Registration Request Letter at the department.*

4.4.5 F0006 – Sample Result Response Letter

28. Description

The department is required to notify the Laboratory of the outcome of the Sample Results contained in the corresponding Sample Result Request Letter. The Sample Result Response Letter is generated at the completion of the capture and processing of the Sample Result Request Letter.

29. Responsibilities

a) Department

1. Process the Sample Result Request Letter
   - Validate that the message complies with specified business rules.
   - Validate that the message complies with the delivery timeframes as specified in the Laboratory appointment.
   - Update department's operational system (ie. AIMS) with the validated data.
2. Generate the Sample Result Response Letter specifying the Outcome of Request Letter.
3. Deliver the Sample Result Response Letter to the Laboratory System messaging service.
4. Track Sample Result Response Letters sent.

b) Laboratory

1. Receive the electronic Sample Result Response Letter.
2. Track Sample Result Response Letters received.
3. Process the Sample Result Response Letter:
   - <Optional> Update Laboratory System with status information.
   - Perform remedial action on unsuccessful information rejected by the department.

30. Message Structure

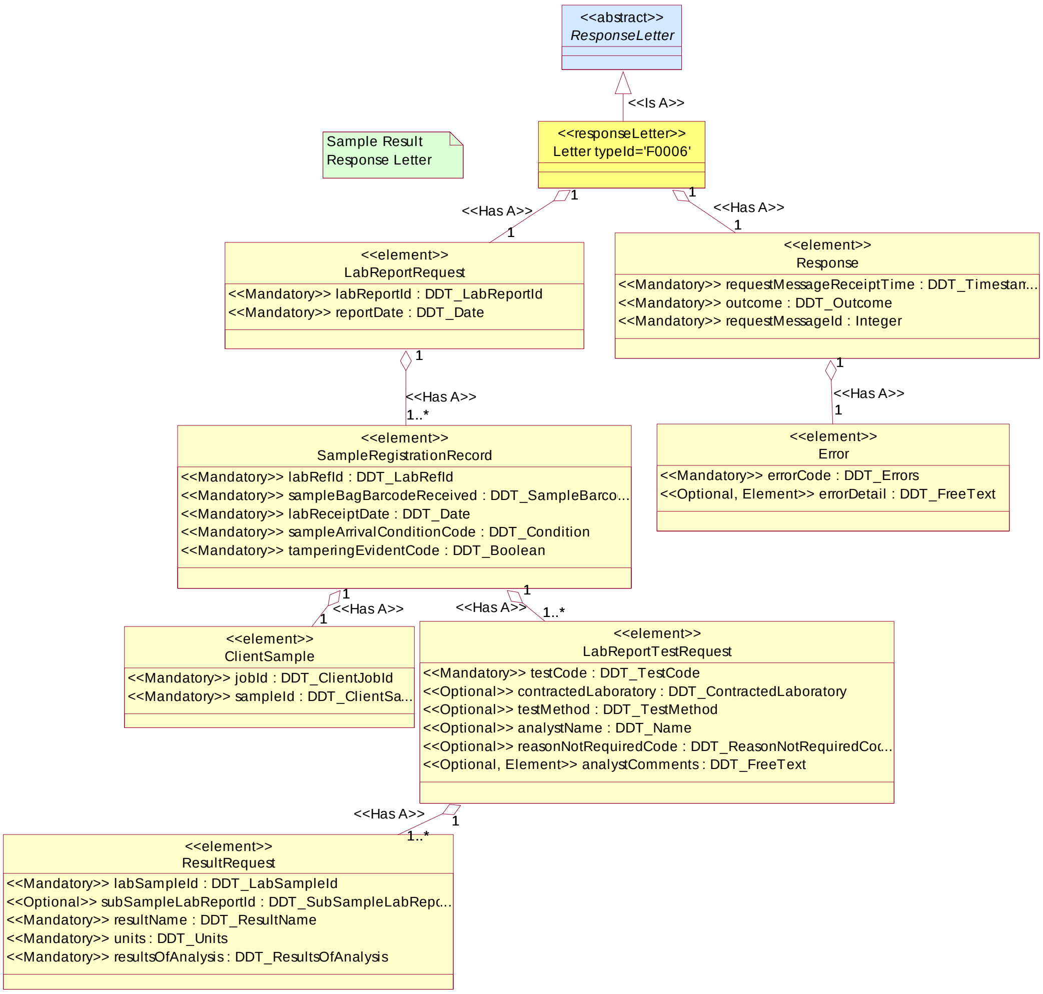

<<abstract>>
ResponseLetter
<<Is A>>
Sample Result
Response Letter
<<responseLetter>>
Letter typeId='F0006'
<<Has A>>
1
1
<<Has A>>
1
1
<<element>>
LabReportRequest
<<Mandatory>> labReportId : DDT_LabReportId
<<Mandatory>> reportDate : DDT_Date
<<element>>
Response
<<Mandatory>> requestMessageReceiptTime : DDT_Timestam...
<<Mandatory>> outcome : DDT_Outcome
<<Mandatory>> requestMessageId : Integer
1
<<Has A>>
1..*
1
<<Has A>>
1
<<element>>
SampleRegistrationRecord
<<Mandatory>> labRefId : DDT_LabRefId
<<Mandatory>> sampleBagBarcodeReceived : DDT_SampleBarco...
<<Mandatory>> labReceiptDate : DDT_Date
<<Mandatory>> sampleArrivalConditionCode : DDT_Condition
<<Mandatory>> tamperingEvidentCode : DDT_Boolean
<<element>>
Error
<<Mandatory>> errorCode : DDT_Errors
<<Optional, Element>> errorDetail : DDT_FreeText
1
<<Has A>>
1
<<Has A>>
1
1..*
<<element>>
ClientSample
<<Mandatory>> jobId : DDT_ClientJobId
<<Mandatory>> sampleId : DDT_ClientSa...
<<element>>
LabReportTestRequest
<<Mandatory>> testCode : DDT_TestCode
<<Optional>> contractedLaboratory : DDT_ContractedLaboratory
<<Optional>> testMethod : DDT_TestMethod
<<Optional>> analystName : DDT_Name
<<Optional>> reasonNotRequiredCode : DDT_ReasonNotRequiredCo...
<<Optional, Element>> analystComments : DDT_FreeText
<<Has A>>
1
1..*
<<element>>
ResultRequest
<<Mandatory>> labSampleId : DDT_LabSampleId
<<Optional>> subSampleLabReportId : DDT_SubSampleLabRep...
<<Mandatory>> resultName : DDT_ResultName
<<Mandatory>> units : DDT_Units
<<Mandatory>> resultsOfAnalysis : DDT_ResultsOfAnalysis

Figure 15 - Sample Result Response Letter Structure

31. Mail Article Specific Rules

1. A LabReportRequest is uniquely identified by [labReportId].
2. One Sample Result Response Letter corresponds to one Laboratory Report.
3. A SampleRegistrationRecord is uniquely identified by [ClientSample/@jobId + ClientSample/@sampleId].
4. A LabReportTestRequest is uniquely identified by [testCode].
5. A ResultRequest is uniquely identified by [labSampleId + resultName].
6. The Response Letter will reference a copy of the original Request Letter - AddressLabel and LabReportRequest elements.
7. If the LabReportRequest is processed without error the Response/@outcome will be set to ‘Pass’ (ie. Sample Result accepted and passed).
8. If the LabReportRequest is processed with an error the Response/@outcome will be set to ‘Fail’ (ie. Sample Result failed) and the appropriate errors must be supplied in the Error element.

32. Assumptions

1. The Envelope is valid.
2. The Response Letter Recipient/@id plus the Response/@requestMessageId uniquely identify the corresponding Request Message.
3. The Response/@requestMessageReceiptTime was created when processing the Request Letter.
4. Those Laboratory Reports that are rejected will need to be resubmitted by the Laboratory in a subsequent message to be processed.

5 Data Definition and Validation Rules

This section describes the attributes and elements that participate in the eResults Mail Articles. This section supplements the business rules defined in Section *4 Mail Article and Business Letter Specifications*.

Refer to the *eResults External Domain Dictionary* for corresponding terminology definitions for XML Entities.

- The first section lists the Data Entities and identifies their eResults Data Type and elements that they participate in.
- The second section defines the eResults Data Types including their type, size, validation / business rules.
- The third section defines some key business rules that apply to dates, message delivery, and other cross-attribute validation / business rules.

5.1 Data Entity Summary

The following table lists the Data Entities and cross-references them to the eResults Mail Article elements they participate in. Refer to *Section 5.2 - Data Type Definitions* for definition and rules for the eResults **'DDT_'** Data Types.

Data Entity	Data Type	XML Elements and Email Subject Element
actualCode	DDT_InspectionCategoryCode	InspectionCategory
address	DDT_Address	ImporterOrAgent
analystComments	DDT_FreeText	LabReportTestRequest
analystName	DDT_Name	LabReportTestRequest
authorisedOfficer	DDT_AuthorisedOfficer	DAFFOffice
brand	DDT_Name	SampleOrderRequest
checksum	DDT_Checksum	Subject
clientReference	DDT_FreeText	References
commercialDescription	DDT_FreeText	SampleOrderRequest
contractedLaboratory	DDT_ContractedLaboratory	LabReportTestRequest
count	DDT_Integer	SampleSize
countryOfOrigin	DDT_CountryISO2Code	Line
createdTimestamp	DDT_Timestamp	AddressLabel
errorCode	DDT_Errors	Error
errorDetail	DDT_FreeText	Error
faxNumber	DDT_PhoneOrFax	DAFFOffice ImporterOrAgent
id	DDT_Integer	MessageIdentification Subject
id	DDT_MessageParty	Recipient Sender Subject (Sender) Subject (Recipient)

Data Entity	Data Type	XML Elements and Email Subject Element
jobId	DDT_ClientJobId	ClientJobRequest ClientSample
labReceiptDate	DDT_Date	SampleRegistrationRecord SampleRegistrationRequest
labRefId	DDT_LabRefId	SampleRegistrationRecord SampleRegistrationRequest
labReportId	DDT_LabReportId	LabReportRequest
labSampleID	DDT_LabSampleId	ResultRequest
lotCode	DDT_LotCode	SampleOrderRequest
mailArticleType		Subject
name	DDT_OfficeName	DAFFOffice
name	DDT_Name	ImporterOrAgent
number	DDT_Line	Line
phoneNumber	DDT_PhoneOrFax	ImporterOrAgent
quantity	DDT_Real	SampleSize
reasonNotRequiredCode	DDT_ReasonNotRequiredCode	LabReportTestRequest
referredCode	DDT_InspectionCategoryCode	InspectionCategory
reportDate	DDT_Date	LabReportRequest
requestMessageReceiptTime	DDT_Timestamp	Response
outcome	DDT_Outcome	Response
resultName	DDT_ResultName	ResultRequest
resultsOfAnalysis	DDT_ResultsOfAnalysis	ResultRequest
sampleArrivalConditionCode	DDT_Condition	SampleRegistrationRecord SampleRegistrationRequest
sampleBagBarcode	DDT_SampleBarcode	SampleOrderRequest
sampleBagBarcodeReceived	DDT_SampleBarcode	SampleRegistrationRecord SampleRegistrationRequest
sampleComments	DDT_FreeText	SampleOrderRequest
sampleDate	DDT_Date	SampleOrderRequest
sampleId	DDT_ClientSample	ClientSample SampleOrderRequest
sampleInspectionConditionCode	DDT_Condition	SampleOrderRequest
sampleTime	DDT_Time	SampleOrderRequest
SubSampleLabReportId	DDT_SubSampleLabReportId	ResultRequest
tamperingEvidentCode	DDT_Boolean	SampleRegistrationRecord SampleRegistrationRequest
testCode	DDT_TestCode	LabReportTestRequest TestRequest
testMethod	DDT_TestMethod	LabReportTestRequest
typeId	DDT_MessageTypeId	Letter

Data Entity	Data Type	XML Elements and Email Subject Element
		MessageIdentification
typeVersion	DDT_Version	MessageIdentification
units	DDT_Units	ResultRequest
unitsCode	DDT_UnitsCode	SampleSize

5.2 Data Type Definitions

The eResults Messaging Service utilises standard Data Types to represent attributes with identical data requirements. The attribute Data Types described below (along with their associated value definitions) must be implemented in the respective XML schema.

Attribute-level Validation Notes that apply to all attributes:

- By default, attributes are not required to be filled to their maximum length (ie. a value is not required for every character of the attribute such as leading zeros) unless otherwise specified in the Attribute-level Validation and Business Rules column.
- Alphanumeric and alphabetic attributes can be supplied in mixed case – coded attributes will be converted to full case during message processing.
- The corresponding Error code is supplied in angle brackets **<Ennnn>** when an Attribute-level Validation or Business Rule is broken.
- Refer to the Format Key at the end of the table for further information of format symbols.

General Attribute-level Validation Rules:

1. **Format** and **Size** must be enforced. **<E0004>**

Data Type	Format	Size (min..max)	Example	Validation and Business Rules
DDT_Address		X(1..140)	Level 10, 20 Allara Street, Canberra ACT 2601	
DDT_AuthorisedOfficer		X(1..10)	ChristieL	2. Must be a valid AIMS Authorised Officer. **<E0100>**
DDT_Boolean	'True' or 'False'	X(4..5)	True	3. Must be from enumerated list. **<E0004>**
DDT_Checksum		X(8..8)	3FC45A0F	4. Invalid Checksum. **<E0005>**
DDT_ClientJobId		X(9..10)	Customs - 1S12920365 Manual - A020001044 - N020021755	5. Must be an existing Entry Number in AIMS. **<E0103>** 6. Entry must not be finalised. **<E0104>**
DDT_ClientSample	NNNA	X(2..4)	1a	7. There must be a corresponding Entry Number

Data Type	Format	Size (min..max)	Example	Validation and Business Rules
				and Sample Id in AIMS. **<E0106>**
DDT_Condition		X(1..5)	TA	8. Must be a valid AIMS Sample Inspection Condition Code (see *eResults Code Requirements)*. **<E0100>**
DDT_ContractedLaboratory		X(1..70)	Paul Newman & Associates	
DDT_CountryISO2Code		X(1..2)	AU	9. Must be a valid AIMS Country Code (see *eResults Code Requirements).* **<E0100>**
DDT_Date	CCYYMMDD	X(8..8)	20020301	10. Must be a valid date. **<E0004>** 11. All timestamps and dates must be not later than the time of reception of the message. **<E0005>**
DDT_Errors	‘[E/I]’ + N(4)	X(5..5)	E0001	12. Must be a valid AIMS Error Code (see *eResults Code Requirements).* **<E0100>** 13. Filled to maximum length. **<E0004>**
DDT_FreeText		X(1..254)		
DDT_InspectionCategoryCode		X(1..2)	IR	14. Must be a valid AIMS Actual Inspection Category Code (see *eResults Code Requirements).* **<E0100>**
DDT_Integer		Integer	25	15. Must be an integer. **<E0004>**
DDT_LabRefId		X(0..50)	VAQIS/02096	
DDT_LabReportId		X(1..15)	RN313361	
DDT_LabSampleId		X(1..30)	N02/028681	
DDT_Line		N(5..5)	00021	16. Filled to maximum length with leading zeros. **<E0004>** 17. Line must not be finalised. **<E0105>** 18. Not negative Integer. **<E0004>**
DDT_LotCode		X(1..30)	1999	
DDT_MessageId		N(0..10)	3445	19. Represents the unique sequential message number for a specific sender (department and Laboratory) of a Request or Response Letter. 20. Not negative Integer. **<E0004>**
DDT_MessageParty		X(1..15)	Laboratory - BERNHTH AQIS - AQIS	21. If a Laboratory then must be a valid AIMS Laboratory Enrolment Identifier Code*.* **<E0100>**

Data Type	Format	Size (min..max)	Example	Validation and Business Rules
DDT_MessageTypeId	ANNNN	X(1..5)	F0003	22. Filled to maximum length. **<E0004>** 23. Must be a valid AIMS eMessage Type Id Code. (see *eResults Code Requirements).* **<E0100>**
DDT_Name		X(1..50)	John Roddick	
DDT_OfficeName		X(1..30)	Sydney	
DDT_Outcome	'Pass' or 'Fail'	X(4..4)	Pass	24. Must be from enumerated list. **<E0004>**
DDT_PhoneOrFax		X(1..20)	02 62725447	
DDT_Real		Real	1.5	
DDT_ResultName		X(1..70)	Carbaryl	
DDT_ResultsOfAnalysis		X(1..50)	<0.10	
DDT_SampleBarcode	'IF' + N(7).	X(9..9)	IF7654321	25. Filled to maximum length with leading zeros. **<E0004>**
DDT_SubSampleLabReportId		X(0..50)	RN2356987	
DDT_TestCode		X(0..8)	ECOLI	26. Must be a valid AIMS Test Code (see *eResults Code Requirements).* **<E0100>** 27. Test results sent to department should correspond to tests requested by department. **<E0102>** 28. There must be a corresponding Entry Number, Sample Id and Test Code in AIMS. **<E0107>**
DDT_TestMethod		X(1..30)	ECOLI	
DDT_Time	HH:MM:SS ±hh:mm	X(14..14)	08:01:32+10:00	29. Must be a valid 24-hour time. **<E0004>**
DDT_Timestamp	YYYY-MM-DDTHH:MM:SS±hh:mm	X(26..26)	2002-02-19T12:19:01 +10:00	30. Must be a valid ISO 9601 extended timestamp format. **<E0004>**
DDT_Units		X(1..20)	Mg/kg	
DDT_Version	NN.NN	X(3..5)	1.0 1.13 2.1	31. Must be a valid AIMS eMessage Type Id Version for the corresponding DDT_MessageTypeId. (see *eResults Code Requirements).* **<E0100>** 32. Mandatory if DDT_MessageTypeId is not null. **<E0009>**
DDT_UnitsCode		X(1..3)	g	33. Must be a valid AIMS Sample Weight Code (see *eResults Code Requirements).* **<E0100>**

FORMAT KEY

Symbol	Meaning
N	Numerical
A	Alphabetic
X	Alphanumeric
CC	Century
YY	Year
MM	Month
DD	Day of month
T	Date / Time Separator
HH	Hours
MM	Minutes
SS	Seconds
sss	Milliseconds
±hh:mm	Offset from Coordinated Universal Time

5.3 General Business Rules

5.3.1 Date Rules

(Sample Date + Sample Time) <= Lab Receipt Date <= Lab Report Date. **<E0101>**

5.3.2 Message Receipt & Delivery Timestamp Rules

Created Timestamp (Request) <= Receipt Timestamp (Response)[1] + 1 Day <= Created Timestamp (Response). **<E0101>**

1. The SQL below shows how 1 day is added to the Receipt Timestamp (Response) – that is the TimeRecieved of the Request Message - to create the Expiry date. That means that no matter when their email arrives, it will not expire for 1 day.

```
CREATE  procedure spAddMailArticle(@MailArticleId uniqueidentifier,  @TimeReceived
datetime, @CreatedTimestamp datetime, @TypeId  varchar(5), @TypeVersion varchar(5),
@SenderId varchar(8))
    as
    begin
        INSERT INTO MailArticle (MailArticleId, TimeReceived, CreatedTimestamp,
    TypeId, TypeVersion,  SenderId, ExpiryDate)
        VALUES (@MailArticleId,  @TimeReceived, @CreatedTimestamp, @TypeId,
    @TypeVersion,  @SenderId, DATEADD(d, 1, @TimeReceived))
    end
GO
```

2. The department will then generate the Created Timestamp (Response) when sending back a Response Message to the Laboratory System who should check that the timestamp sequence is in order (ie. Greater than their Created Timestamp (Request)).

[1] This is actually the timestamp that the Request message was received and saved.

3. When the department start sending eOrder messages, the department will be required to check the timestamps on the response messages sent from the Laboratory System to ensure that the timestamp sequence is in order.

5.3.3 Other Multiple-Attribute Validation Rules

If Sample Quantity or Sample Weight Code is not NULL then both values must be not NULL. **<E0009>**

5.3.4 Mandatory Rules

Each Data Entity referenced in a mail article will specify a mandatory rule which are coded as:

- Mandatory
- Optional

5.4 Error Processing

eResults Messaging notifies Laboratories of errors in processing Mail Articles by returning Acknowledgements or Response Letters in the form of XML documents. Both documents have a Response element indicating an Outcome of 'Fail' and containing an Error element.

Errors occurring in the processing of Mail Articles consist of an Error Code and Error Detail.

All possible processing errors have been assigned an Error Code with the format 'Ennnn', where n is a digit (0-9). General descriptions have been assigned to each error as the Error Detail. Where possible, a more specific description of an error is given in the Error Detail. An example of this is parsing errors returned by the MSXML DOM when an XML Document is being checked to confirm it is well-formed and conforms to the relevant schema.

Errors ranging from E0001 to E0099 are concerned with Certified Mail delivery and general validation. Errors ranging from E0100 to E9999 are concerned with specific Business Letter validation and business rules.

6 Operational Considerations

6.1 Maintaining Code Sets Synchronisation

It is essential that all Laboratory Systems and the department have eResults code sets synchronised to avoid message failures resulting from unidentified codes.

The following scenarios require code synchronisation:

- Codes that are required to be interpreted by Laboratories as instructions (eg. ordering a specific test).
- Codes that are required to understand the success of Request Letters sent by the Laboratory System (via interpreting a Response Letter).
- Codes that form information that informs the department of a service being completed.

Due to the small number of trading partners and code sets the method for synchronising codes will be achieved by the department manually issuing emails with updates to Laboratories. The email will have the code table(s) attached that consists of the code, description and whether it is a new code, modified description or inactivated code. This may be imported electronically in to the Laboratory System or modified manually.

Refer to the *eResults External Code Requirements* document for a full description on eResults codes*.*

6.1.1 External Codes to Synchronise

a) Message Handling

Both the department and Laboratory Systems are required to process and generate these codes.

1. Errors.
2. Laboratory Identifier (One Value).
3. Message Type
    a. Message Type Id.
    b. Message Type Version.

b) Other Request Letter Codes

These are codes that the Laboratory System must create as output information in a Request Letter.

4. Sample Inspection Condition Code.
5. Test Code.

6.1.2 Department Internal Codes

These are other codes that the the department's System must create as output information in a Request Letter (and returned by Laboratory Systems in a Response Letter). These codes will not require synchronisation or validation in Laboratory Systems, but a rule will be implemented ensuring values sent to Laboratory Systems have not been modified by the Laboratory Systems in the Response Letter.

1. Actual Inspection Category Code.
2. Authorised Officer.
3. Office Code.
4. Country of Origin.

5. Referred Inspection Category Code.
6. Sample Inspection Condition Code.
7. Sample Unit Code.
8. Test Code.

6.2 Amending Information

There is always the need to modify information, which may result from erroneous lodgement in the first instance, or simply because a situation has changed (eg. An importer elects to re-export or destroy the food instead of testing).

The system will not support amendments to information that has been received and processed. The system will support additional information being supplied in new messages to augment previously submitted information. Deleting information is not allowed but may be deactivated manually. The need for the deletion feature will be monitored and implemented only if required in the future.

Some amendments will be implemented with an operational procedure that will require manual intervention involving both the department and Laboratories. For example, if tests were no longer required for a line but already ordered, the department will update the AIMS System and set tests to 'not required', provide a reason and apply the appropriate final direction. The Laboratory will also need to update their Laboratory System to reflect this change.

If the System detects that information arriving in a message already exists it will do one of two things:

- If the data matches the message it will process the message immediately.
- If the data does not match the message will be pended for human intervention. In this scenario the Laboratory System will still receive a Response Letter (Pass or Fail). The operator may then choose to:
    - Apply the message content.
    - Discard the message content.

6.3 Help Desk and Support

The department will develop and distribute a set of guidelines for resolving likely interface problems that will encompass both the department and laboratories perspectives. Additional in-house procedures may be developed by laboratories to compliment these guidelines. The ability to revert back to manual procedures and data entry will be essential in cases of extreme eResults failures.

The following list summarises the type of message failure events that will be supported.

1. Message sent by Laboratory System but not received by the department.
2. Message sent by the department but not received by the Laboratory System.
3. Importer System does not receive the e-Advice message from the department.
4. Message sent by Laboratory System delivered to the department out of sequence.
5. The same Laboratory System message is delivered to the department more than once.
6. Message sent by Laboratory System corrupted in transmission to the department, resulting in an inability to parse and process the message.
7. Message sent by the department corrupted in transmission to Laboratory System, resulting in an inability to parse and process the message.

8. Message sent by Laboratory System corrupted in transmission to the department, the system is able to parse and the system is able to process the message.
9. Message sent by Laboratory System contains incorrect data.
10. Department e-mail system outage (availability).
11. Laboratory e-mail system outage (availability).
12. Malicious user floods the department system with e-mail (availability).
13. Catastrophic failure of the department e-mail system leading to loss of e-mails (data security).
14. Catastrophic failure of Laboratory e-mail system leading to loss of e-mails (data security).
15. Persons intercept e-mail traffic and distributes/uses information (confidentiality).
16. Person impersonates the department in generating e-mail message (sender authentication).
17. Person impersonates Laboratory System in generating e-mail message (sender authentication).

The following items summarise other type of events that will be supported.

18. Amending information already sent (ie. adding, changing and deleting information) messages.
19. Receiving information in a message that has information already entered (either from a previous message or manual input).

For Information:

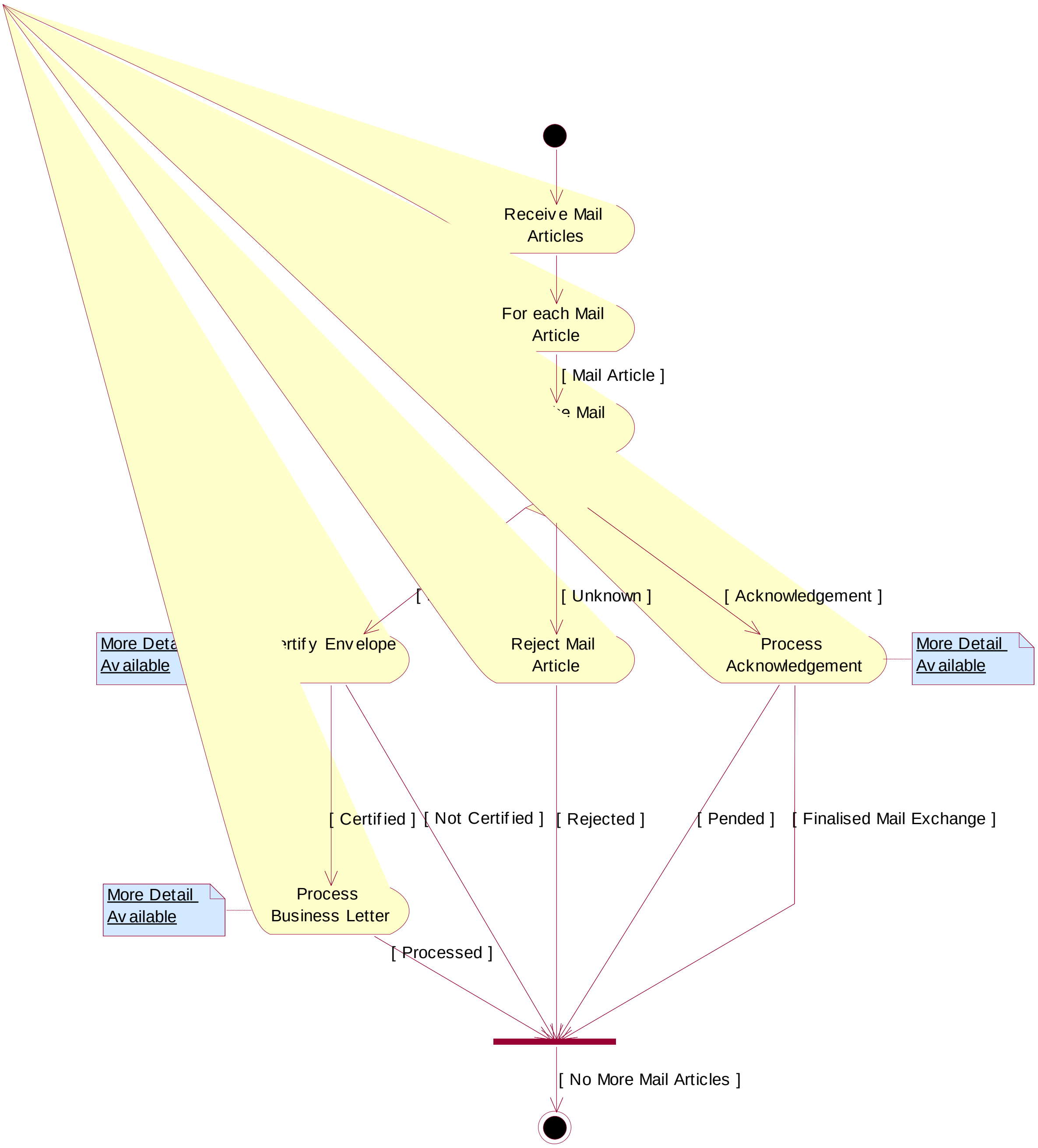
The department will develop a Message Queue Management Tool to assist support activities and will provide the following key features:

1. *Search for delivered and received messages using Address Label parameters, Processing Status and Outcome.*
2. *Display a message in a viewer displaying its field tag and value sets.*
3. *Allow secondary string searching on a selected message to refine the search for a given message.*
4. *Display related message(s) for a selected message: eg. For an Envelope/Request Letter display the - Acknowledgement of the Envelope/Request Letter, Envelope/Response Letter, Acknowledgment of the Envelope/Response Letter.*
5. *Manually pass a message (ie. generates and posts the Pass Response Letter but does not apply the data in the message).*
6. *Manually Fail a message (ie. generates and posts the Fail Response Letter but does not apply the data in the message).*
7. *(Re)Apply Message (ie. generates and posts the Pass Response Letter and applies the data in the message replacing existing data in the System).*
8. *Resend Message (ie, resends a message already sent – does not change the message but records the event).*
9. *Discard Message (ie. marks the message as discarded with no further processing).*
10. *Navigate to an AIMS Entry from the Message Queue Management Tool.*

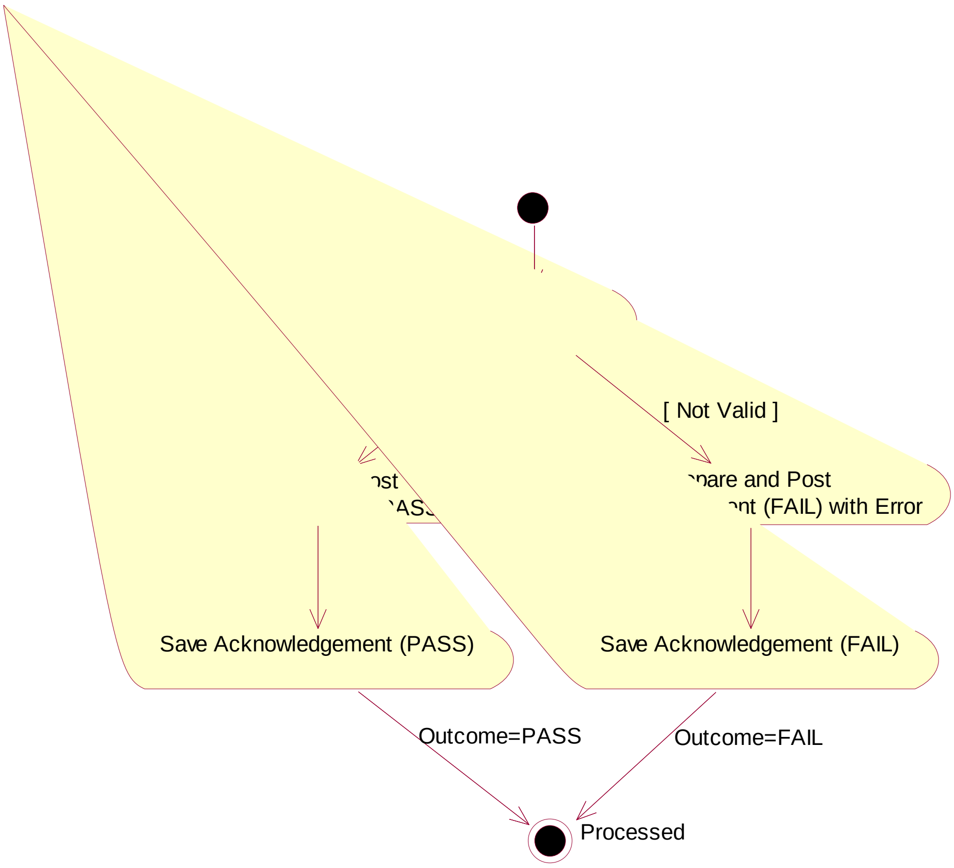
Appendix A – Activity Diagrams

This section includes activity diagrams that pictorially represent the eResults flow of events described in Section *3.3 Messaging Process*.

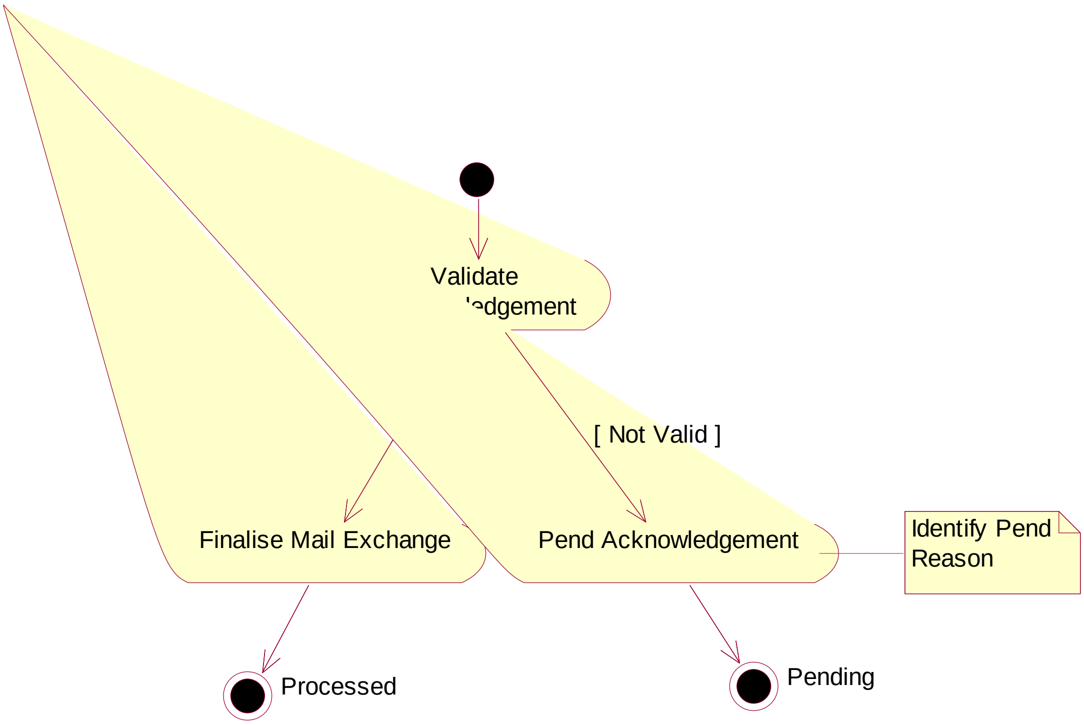
Main Flow of Events



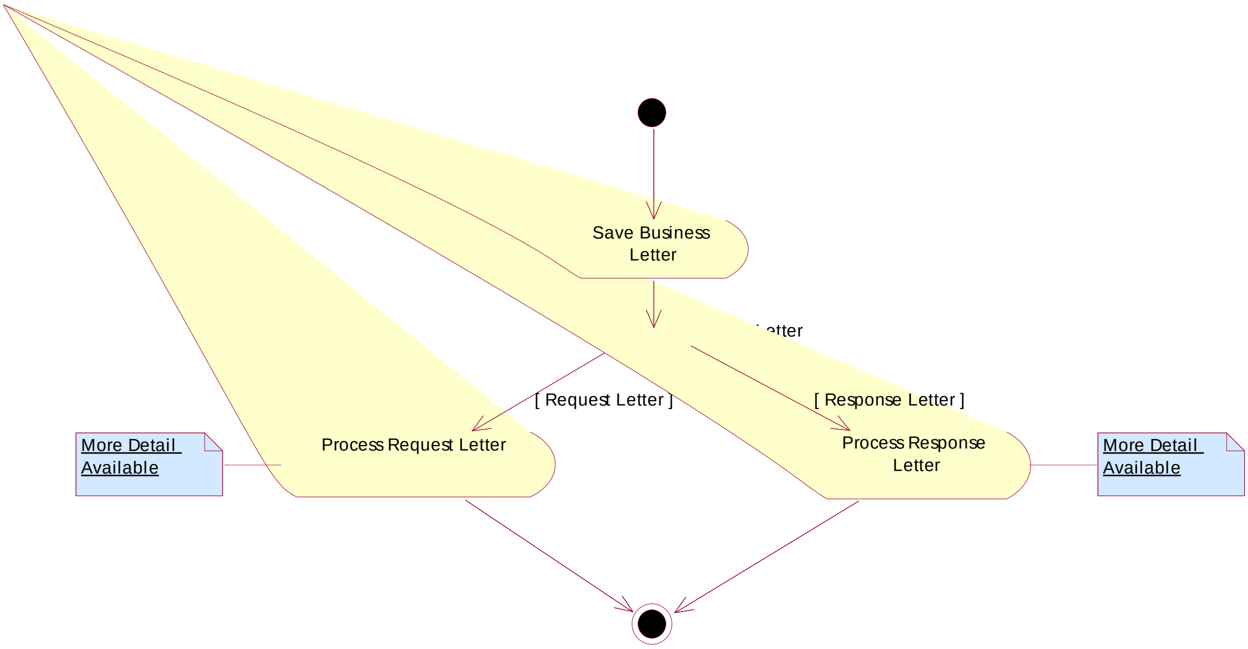
Certify Envelope



Process Acknowledgement

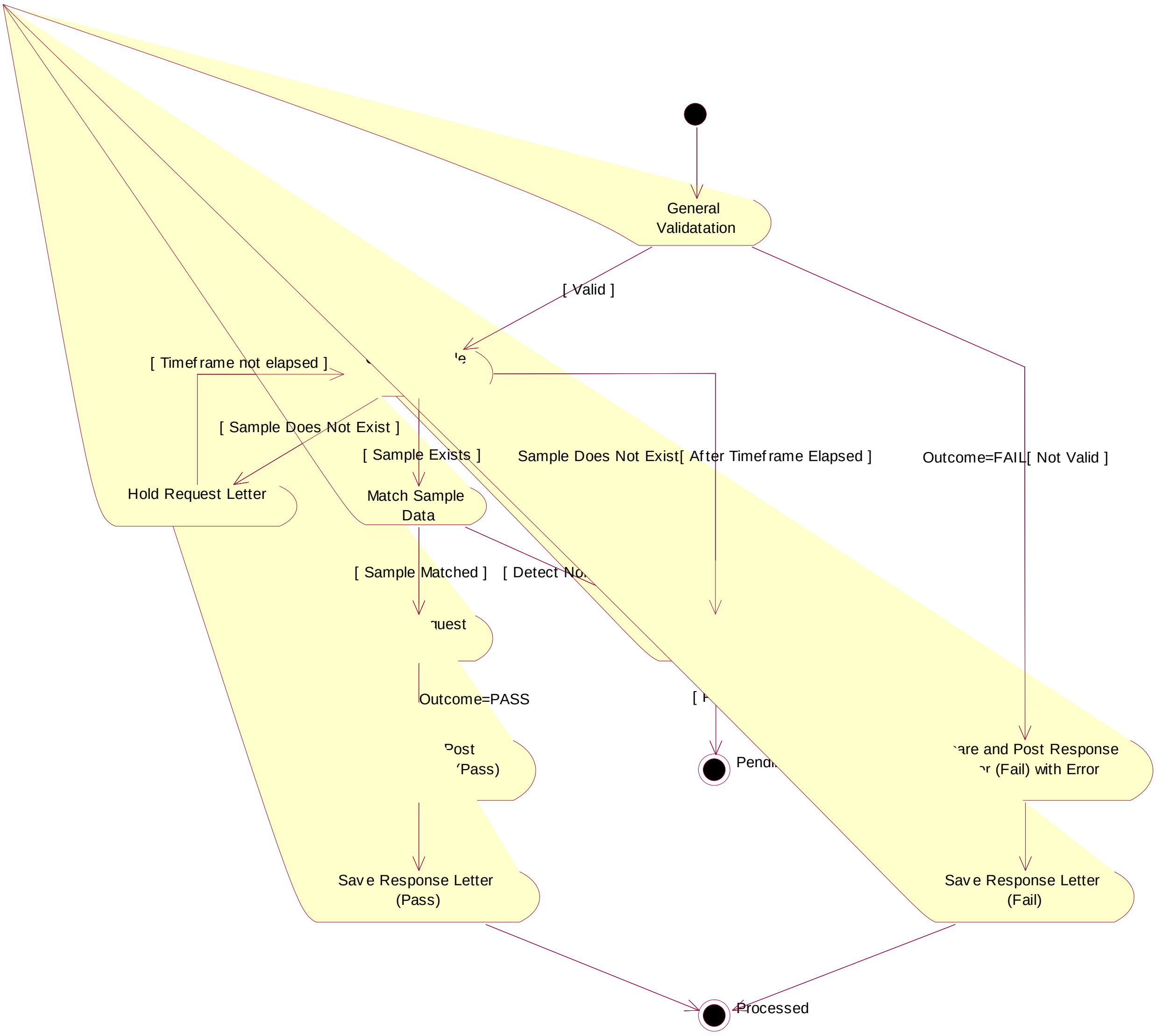


Process Business Letter



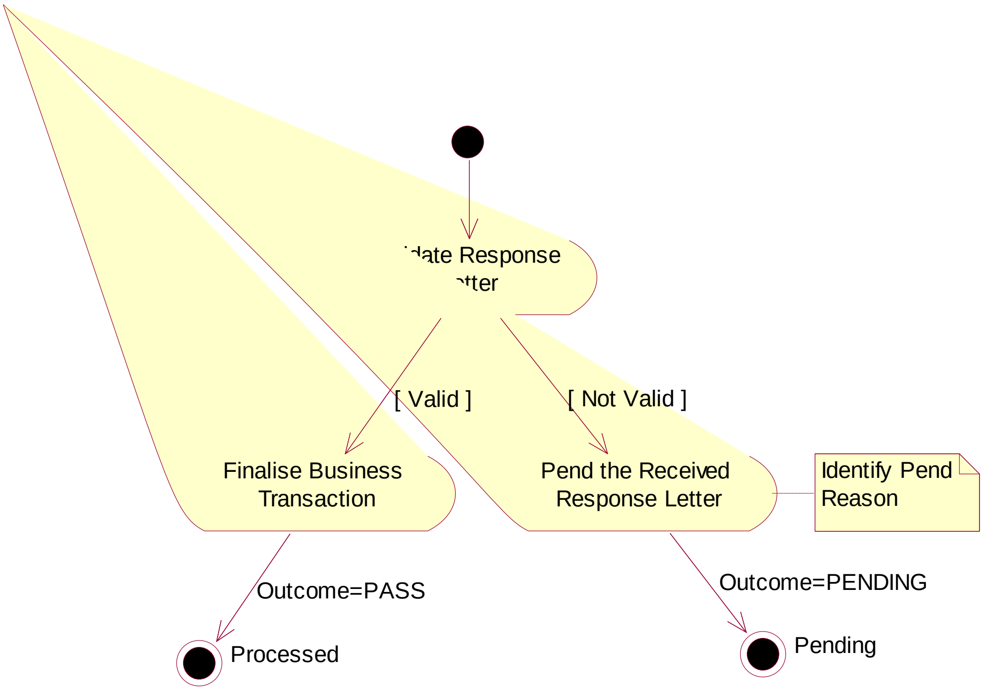
Process Request Letter

This flow is specific to the department processing a Request Letter.

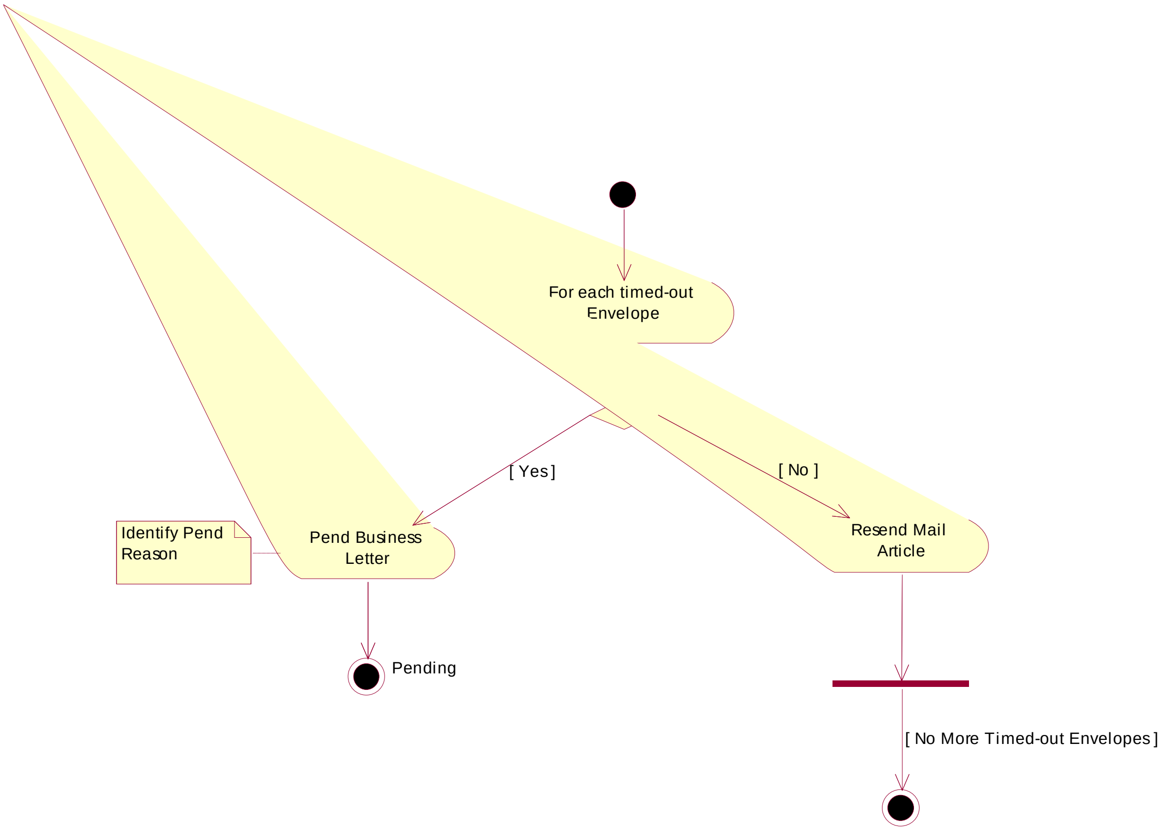


Process Response Letter

This flow is specific to the department processing a Response Letter.



Envelope Not Acknowledged

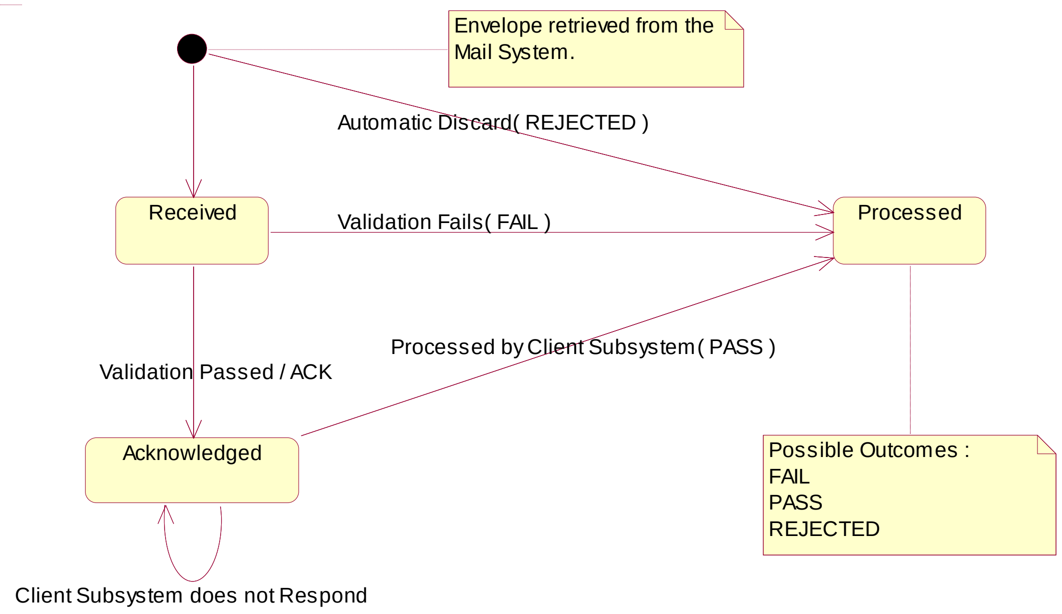


Appendix B – State Transition Diagrams

This appendix describes the State Transition of Mail Articles in the Certified Mail System (CMS) and the State Transition of Business Letters adopted by the department. It also illustrates Outcomes of particular States where applicable.

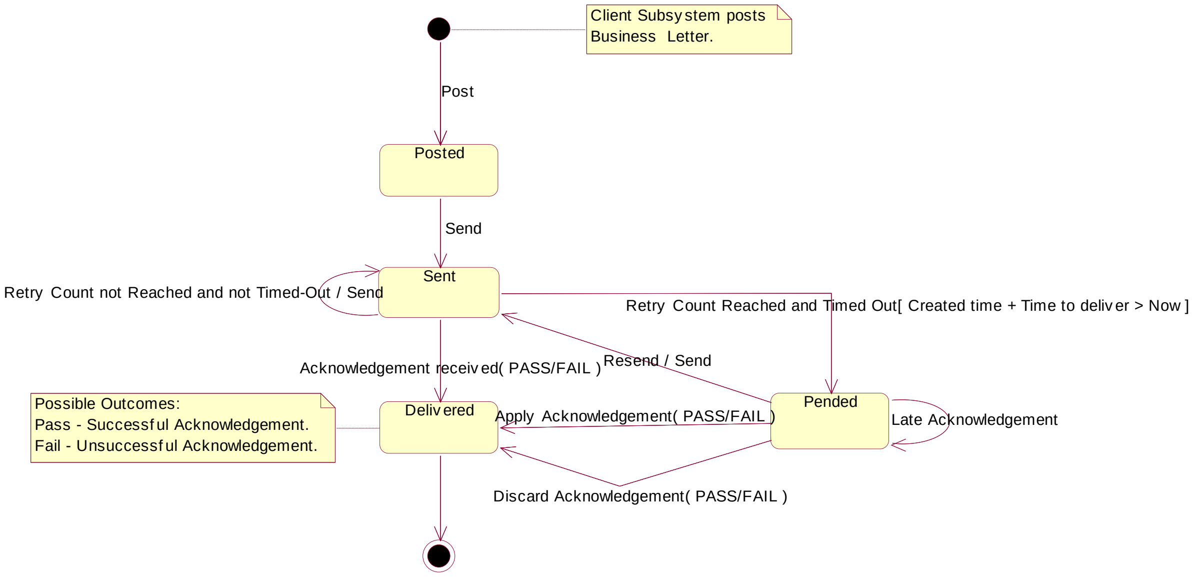
CMS Envelopes Received (State Transitions)

This diagram illustrates the State transitions for Envelopes received (and their Acknowledgement) by the department.


Envelope retrieved from the Mail System.
Automatic Discard(REJECTED)
Received
Validation Fails(FAIL)
Processed
Processed by Client Subsystem (PASS)
Validation Passed / ACK
Acknowledged
Possible Outcomes :
FAIL
PASS
REJECTED
Client Subsystem does not Respond

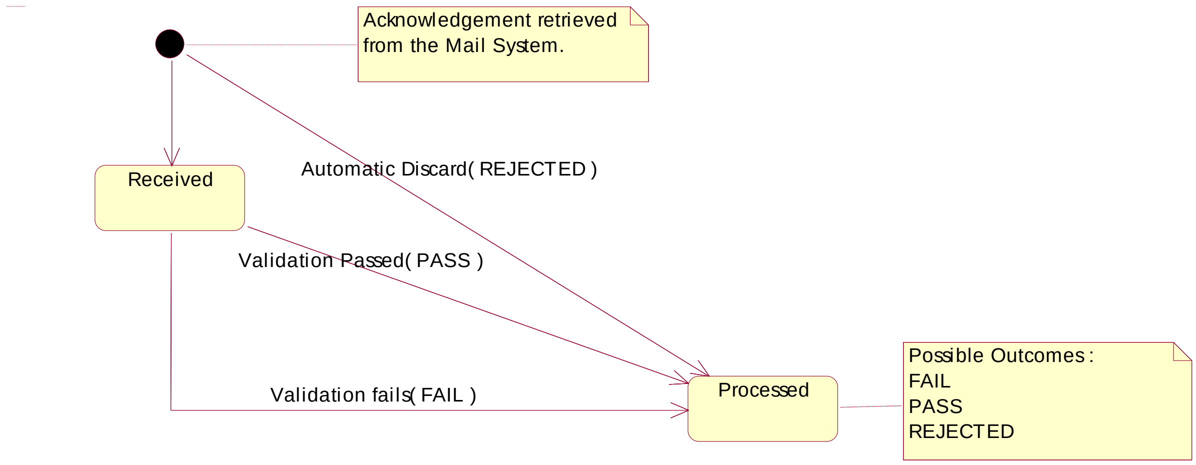
CMS Envelopes Sent (State Transitions)

This diagram illustrates the State transitions for Envelopes sent (and their Acknowledgement) by the department.


Client Subsystem posts Business Letter.
Post
Posted
Send
Retry Count not Reached and not Timed-Out / Send
Sent
Retry Count Reached and Timed Out[Created time + Time to deliver > Now]
Acknowledgement received(PASS/FAIL)
Resend / Send
Possible Outcomes:
Pass - Successful Acknowledgement.
Fail - Unsuccessful Acknowledgement.
Delivered
Apply Acknowledgement(PASS/FAIL)
Pended
Late Acknowledgement
Discard Acknowledgement(PASS/FAIL)

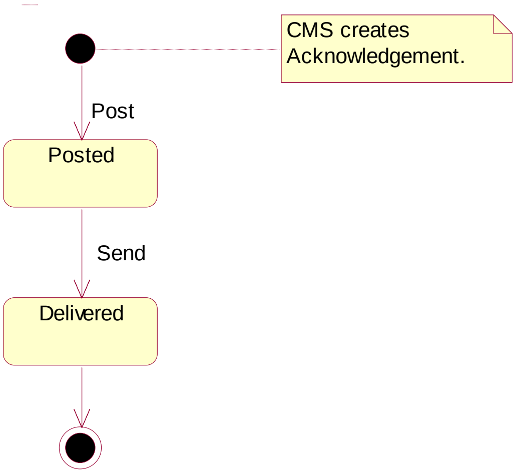
CMS Acknowledgments Received (State Transitions)

This diagram illustrates the State transitions for Acknowledgments received by the department.


Acknowledgement retrieved from the Mail System.
Received
Automatic Discard(REJECTED)
Validation Passed(PASS)
Validation fails(FAIL)
Processed
Possible Outcomes :
FAIL
PASS
REJECTED

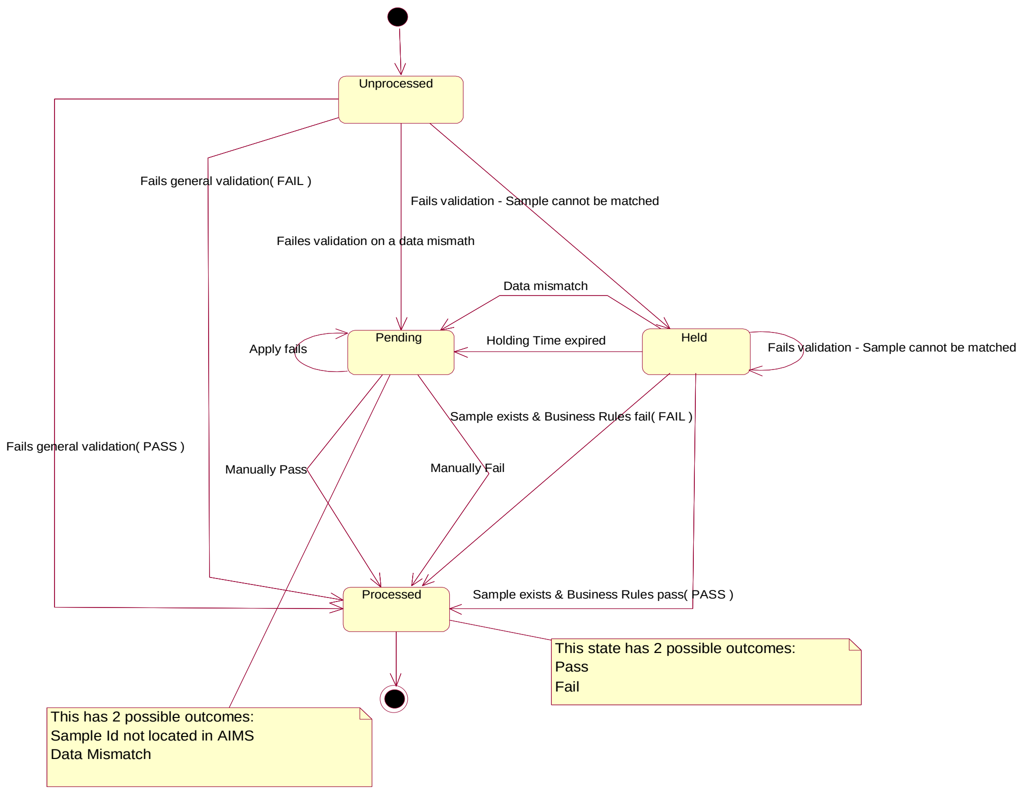
CMS Acknowledgments Sent (State Transitions)

This diagram illustrates the State transitions for Acknowledgments received by the department.



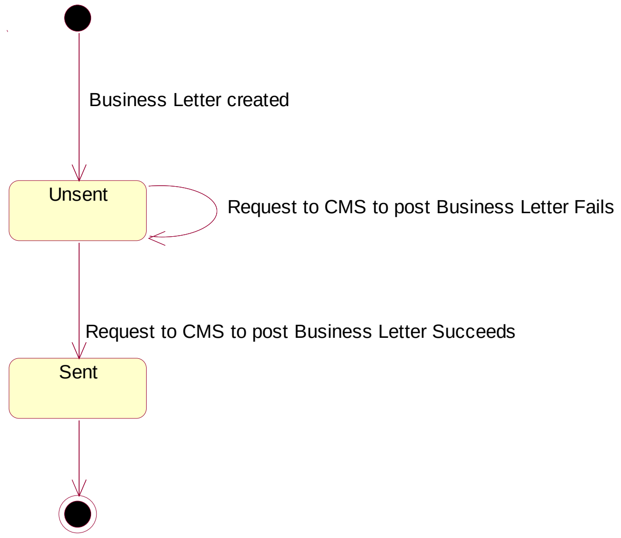
eResults Business Letters Received (State Transitions)

This diagram illustrates the State transitions for Business Letters received by the department.


Unprocessed
Fails general validation(FAIL)
Fails validation - Sample cannot be matched
Failes validation on a data mismath
Data mismatch
Apply fails
Pending
Holding Time expired
Held
Fails validation - Sample cannot be matched
Sample exists & Business Rules fail(FAIL)
Fails general validation(PASS)
Manually Pass
Manually Fail
Processed
Sample exists & Business Rules pass(PASS)
This state has 2 possible outcomes:
Pass
Fail
This has 2 possible outcomes:
Sample Id not located in AIMS
Data Mismatch

eResults Business Letters Sent (State Transitions)

This diagram illustrates the State transitions for Business Letters sent by the department.

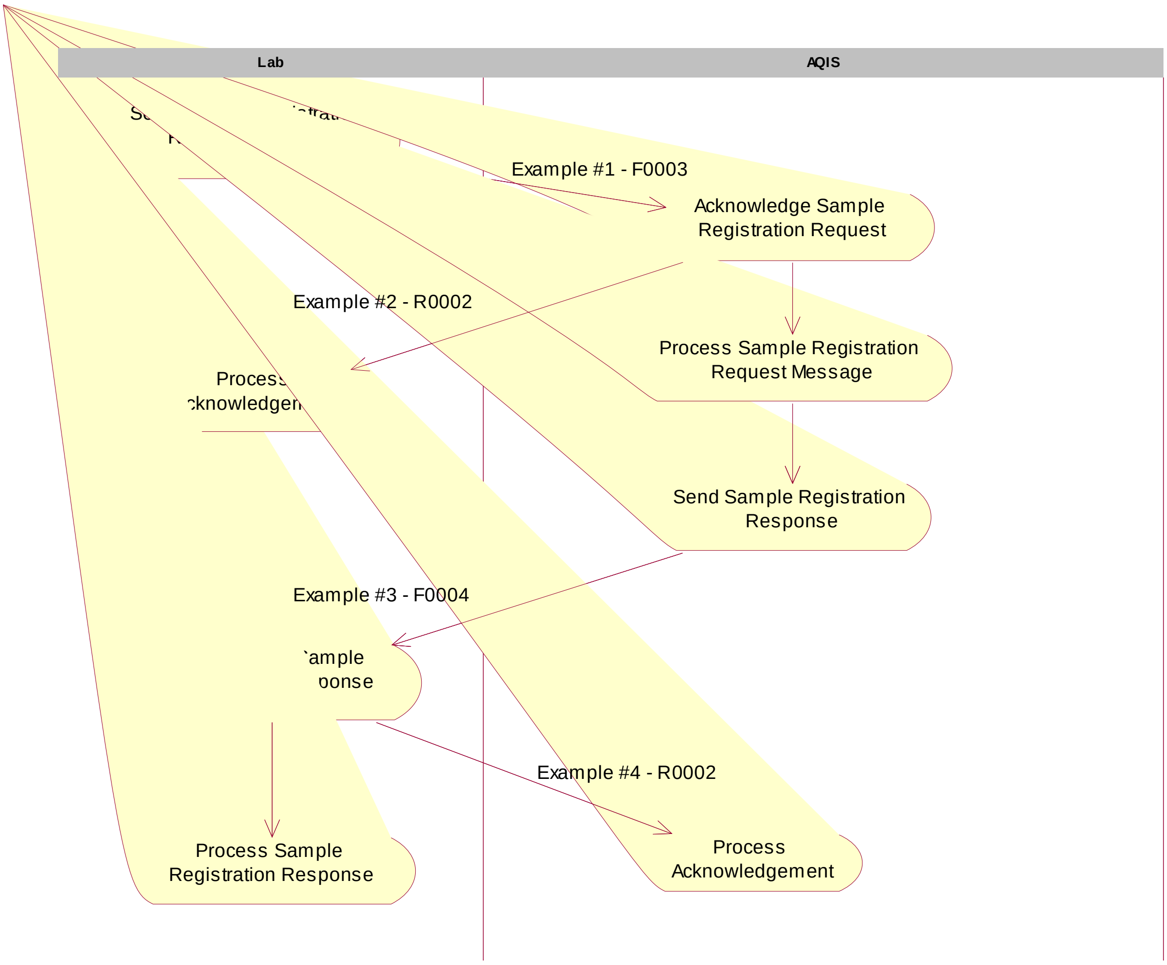

Business Letter created
Unsent
Request to CMS to post Business Letter Fails
Request to CMS to post Business Letter Succeeds
Sent

Appendix C – XML Examples

This section will provide examples of each of the eResults Message Types.

Successful end-to-end Sample Registration delivery process

This example describes the successful end-to-end Sample Registration delivery, acknowledgement and response process.


Lab
AQIS
Example #1 - F0003
Acknowledge Sample Registration Request
Example #2 - R0002
Process Sample Registration Request Message
Process
Send Sample Registration Response
Example #3 - F0004
Example #4 - R0002
Process Sample Registration Response
Process Acknowledgement

a) Example 1 – Sample Registration Request (F0003)

This is an example of a Sample Registration Request Message being sent from a Laboratory System to the department.

```
<!-- This is an example of a Laboratory System sending an F0003 Sample
Registration Request to the department.-->
<!--
 This example Envelope shows that a F0003 Sample Registration Request
Letter
 is being sent from the BERS = Bernard Heath Laboratory System to
eResults.
 -->
<Envelope>
 <AddressLabel createdTimestamp="2001-12-17T09:30:47-05:00">
  <MessageIdentification id="10" typeId="F0003" typeVersion="1.0"/>
  <Sender id="BERS"/>
  <Recipient id="eResults"/>
 </AddressLabel>
 <!-- Contained within the BusinessContent segment is the actual Business
Letter -->
 <BusinessContent>
  <!--
    In this example it is the Sample Registration Request Message
(F0003).
   -->
  <Letter typeId="F0003">
   <!-- This is an example of a Sample Registration for Entry Number
3B12349876 and Sample Id 1a. -->
   <SampleRegistrationRequest labRefId="058239"
sampleBagBarCodeReceived="IF7654321" labReceiptDate="2003-01-29"
sampleArrivalCondition="TA" tamperingEvidentCode="False">
    <ClientSample jobId="3B12349876" sampleId="1a"/>
   </SampleRegistrationRequest>
  </Letter>
 </BusinessContent>
</Envelope>
```

b) Example 2 – Acknowledgment (PASS)

This is an example of the department acknowledging the receipt of a Sample Registration Request Message sent from a Laboratory System.

```
<!--
 This is an example of the department’s Certified Mail System (CMS)
acknowledging the receipt of an envelope
 containing the F0003 Sample Registration Request Letter on behalf of
eResults.
 -->
<!--
 The Acknowledgement is standard for all eResults messages and
acknowledges the receipt of an Envelope.
 It contains an Address Label that is structurally identical to the
Envelope Address Label.
-->
<Acknowledgement>
 <AddressLabel createdTimestamp="2003-01-29T09:10:00">
  <MessageIdentification id="100" typeId="R0002" typeVersion="1.0"/>
  <Sender id="eResults"/>
  <Recipient id="BERS"/>
 </AddressLabel>
 <!--
  The Response indicates whether the envelope is acknowledged (Pass) or
not acknowledged (Fail).
  In this example the envelope is acknowledged with a Pass.
  -->
 <Response requestMessageId="10" requestMessageReceiptTime="2003-01-
29T09:05:00" outcome="Pass"/>
</Acknowledgement>
```

c) Example 3 – Sample Registration Response (F0004)

This is an example of department’s sending a Sample Registration Response Message (in response to a Sample Registration Request Message) to a Laboratory System.

```
<!--
 This is an example of the eResults System responding to the Registration
Request message.
 It verifies that the Business Letter content is well formed XML and
passes all business rules.
 -->
<!--
 This example Envelope shows that a F0004 Sample Registration Response
Letter is being sent
 from the the department to the BERS = Bernard Heath Laboratory System.
-->
<Envelope>
 <AddressLabel createdTimestamp="2003-01-29T09:14:00">
  <MessageIdentification id="101" typeId="F0004" typeVersion="1.0"/>
  <Sender id="eResults"/>
  <Recipient id="BERS"/>
 </AddressLabel>
 <BusinessContent>
  <!--
   In this example it is the Sample Registration Response Message
(F0004).
   -->
  <Letter typeId="F0004">
   <!-- Includes the original sample registration information for Entry
Number 3B12349876 and Sample Id 1a. -->
   <SampleRegistrationRequest labRefId="058239"
sampleBagBarCodeReceived="IF7654321" labReceiptDate="2003-01-29"
sampleArrivalCondition="TA" tamperingEvidentCode="False">
    <ClientSample jobId="3B12349876" sampleId="1a"/>
   </SampleRegistrationRequest>
   <Response requestMessageId="10" requestMessageReceiptTime="2003-01-
29T09:13:00" outcome="Pass"/>
  </Letter>
 </BusinessContent>
</Envelope>
```

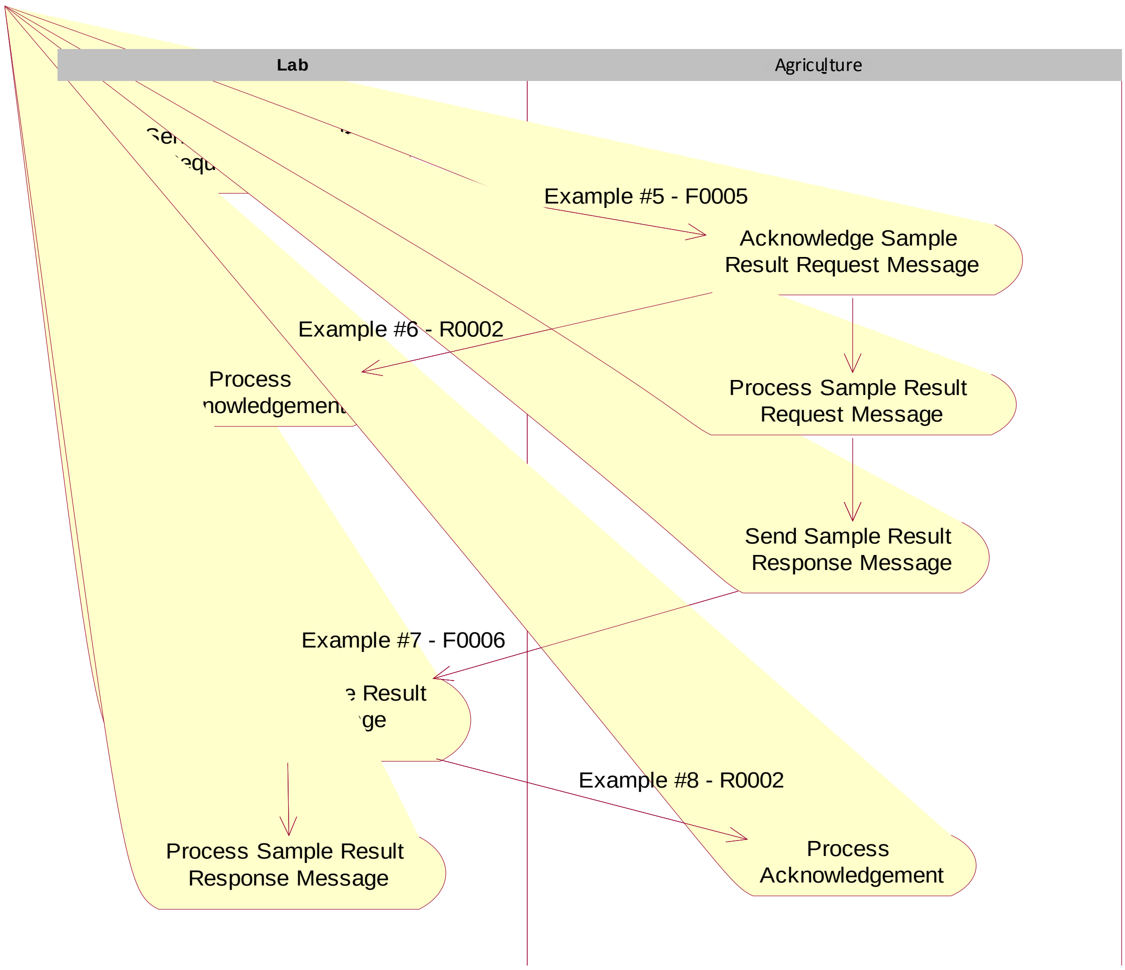
d) Example 4 – Acknowledgment (PASS)

This is an example of a Laboratory System acknowledging the receipt of a Sample Registration Response Message sent from the department.

```
<!--
 This is an example of the Laboratory Certified Mail System (CMS)
 acknowledging the receipt of the envelope containing the  F0004 Sample
Registration Response Letter sent by the department.
 -->
<!--
 The Acknowledgement is standard for all eResults messages and
acknowledges the receipt of an Envelope.
 It contains an Address Label that is structurally identical to the
Envelope Address Label.
-->
<Acknowledgement>
 <AddressLabel createdTimestamp="2003-01-29T09:15:00">
  <MessageIdentification id="11" typeId="R0002" typeVersion="1.0"/>
  <Sender id="BERS"/>
  <Recipient id="eResults"/>
 </AddressLabel>
 <!--
  The Response indicates whether the envelope is acknowledged (Pass) or
not acknowledged (Fail).
  In this example the envelope is acknowledged.
  -->
 <Response requestMessageId="101" requestMessageReceiptTime="2003-01-
29T09:14:50" outcome="Pass"/>
</Acknowledgement>
```

Successful end-to-end Sample Result delivery process

This example describes the successful end-to-end Sample Result delivery, acknowledgement and response process.


Lab
Agriculture
Ser
equ
Example #5 - F0005
Acknowledge Sample Result Request Message
Example #6 - R0002
Process
nowledgement
Process Sample Result Request Message
Send Sample Result Response Message
Example #7 - F0006
e Result
ge
Example #8 - R0002
Process Sample Result Response Message
Process Acknowledgement

e) Example 5 – Sample Result Request (F0005)

This is an example of a Sample Result Request Message being sent from a Laboratory System to the department.

```
<!--
 This is an example of a Laboratory System sending an F0005 Sample Result
Request to the department.
-->
<!--
 This example Envelope shows that a F0005 Sample Result Request Letter
 is being sent from the BERS = Bernard Heath Laboratory System to
eResults.
-->
<Envelope>
 <AddressLabel createdTimestamp="2003-01-30T09:00:00">
  <MessageIdentification id="12" typeId="F0005" typeVersion="1.0"/>
  <Sender id="BERS"/>
  <Recipient id="eResults"/>
 </AddressLabel>
 <BusinessContent>
  <!-- Contained within the Letter segment is the actual Business Letter
-->
  <Letter typeId="F0005">
   <!-- This example shows a Bernard Heath Report RN313361 dated 20
March 2003. -->
   <LabReportRequest labReportId="RN313361" reportDate="2003-01-30">
    <!--
    This is an example of a Sample Registration for Entry Number
3B12349876 and Sample Id 1a.
    -->
    <SampleRegistrationRecord labRefId="058239"
sampleBagBarCodeReceived="IF7654321" labReceiptDate="2003-01-29"
sampleArrivalCondition="TA" tamperingEvidentCode="False">
     <ClientSample jobId="3B12349876" sampleId="1a"/>
     <!--
      Each Sample can have multiple tests requested against it.
      This example shows multiple Tests for the Sample.
      -->
     <!-- Single result test -->
     <LabReportTestRequest testCode="AFLAT" contractedLaboratory="NMI"
testMethod="AFLATOX" analystName="John Smith">
      <AnalystComments/>
      <!-- A single result test will have only one result. -->
      <ResultRequest labSampleId="N02/028681" resultName="AFLAT"
units="µg/kg" resultsOfAnalysis="&lt;5"/>
     </LabReportTestRequest>
     <!--
      Merged-result tests

      Following five tests, ECOLI, SPC, Salmonella, Staph Enterotoxin
and Vibrio Cholera are usually performed together,
      constituting the standard department microbiologicals set of tests
      -->
     <!-- First Microbiological test -->
     <LabReportTestRequest testCode="ECOLI" contractedLaboratory=""
testMethod="ECOLI" analystName="Brian Brown">
      <AnalystComments/>
      <!-- Each Test can have multiple results.  This example shows FIVE
RESULTS for the ECOLI Test. -->
      <ResultRequest labSampleId="N02/028681" resultName="ECOLI1"
units="cfu/g" resultsOfAnalysis="&lt;2.3"/>
      <ResultRequest labSampleId="N02/028681" resultName="ECOLI2"
units="cfu/g" resultsOfAnalysis="&lt;2.3"/>
```

```
       <ResultRequest labSampleId="N02/028681" resultName="ECOLI3"
units="cfu/g" resultsOfAnalysis="&lt;189.0"/>
       <ResultRequest labSampleId="N02/028681" resultName="ECOLI4"
units="cfu/g" resultsOfAnalysis="&lt;6.0"/>
       <ResultRequest labSampleId="N02/028681" resultName="ECOLI5"
units="cfu/g" resultsOfAnalysis="&lt;2.3"/>
      </LabReportTestRequest>
      <!-- Second Microbiological test -->
      <LabReportTestRequest testCode="SPC" contractedLaboratory=""
testMethod="SPC" analystName="Brian Brown">
       <AnalystComments/>
       <!-- Each Test can have multiple results.  This example shows FIVE
RESULTS for the SPC Test. -->
       <ResultRequest labSampleId="N02/028681" resultName="SPC1"
units="cfu/g" resultsOfAnalysis="10.4E3"/>
       <ResultRequest labSampleId="N02/028681" resultName="SPC2"
units="cfu/g" resultsOfAnalysis="2.6E4"/>
       <ResultRequest labSampleId="N02/028681" resultName="SPC3"
units="cfu/g" resultsOfAnalysis="2.4E4"/>
       <ResultRequest labSampleId="N02/028681" resultName="SPC4"
units="cfu/g" resultsOfAnalysis="1.7E3"/>
       <ResultRequest labSampleId="N02/028681" resultName="SPC5"
units="cfu/g" resultsOfAnalysis="3.5E5"/>
      </LabReportTestRequest>
      <!-- Third Microbiological test -->
      <LabReportTestRequest testCode="SALM" contractedLaboratory=""
testMethod="SALMONELLA" analystName="Brian      Brown">
       <AnalystComments/>
       <!-- Each Test can have multiple results.  This example shows FIVE
RESULTS for the SALMONELLA Test. -->
       <ResultRequest labSampleId="N02/028681"
subSampleLabReportId="RN313362" resultName="SALM1" units="cfu/25g"
resultsOfAnalysis="1"/>
       <ResultRequest labSampleId="N02/028681"
subSampleLabReportId="RN313363" resultName="SALM2" units="cfu/25g"
resultsOfAnalysis="1"/>
       <ResultRequest labSampleId="N02/028681"
subSampleLabReportId="RN313364" resultName="SALM3" units="cfu/25g"
resultsOfAnalysis="1"/>
       <ResultRequest labSampleId="N02/028681"
subSampleLabReportId="RN313365" resultName="SALM4" units="cfu/25g"
resultsOfAnalysis="2"/>
       <ResultRequest labSampleId="N02/028681"
subSampleLabReportId="RN313366" resultName="SALM5" units="cfu/25g"
resultsOfAnalysis="1"/>
      </LabReportTestRequest>
      <!-- Fourth Microbiological test -->
      <LabReportTestRequest testCode="STAPE" contractedLaboratory=""
testMethod="Staph Enterotoxin" analystName="Brian Brown">
       <AnalystComments/>
       <!-- Each Test can have multiple results.  This example shows FIVE
RESULTS for the Staph Enterotoxin Test. -->
       <ResultRequest labSampleId="N02/028681" resultName="STAPE1"
units="" resultsOfAnalysis="ND"/>
       <ResultRequest labSampleId="N02/028681" resultName="STAPE2"
units="" resultsOfAnalysis="ND"/>,
       <ResultRequest labSampleId="N02/028681" resultName="STAPE3"
units="" resultsOfAnalysis="ND"/>
       <ResultRequest labSampleId="N02/028681" resultName="STAPE4"
units="" resultsOfAnalysis="ND"/>
       <ResultRequest labSampleId="N02/028681" resultName="STAPE5"
units="" resultsOfAnalysis="ND"/>
      </LabReportTestRequest>
      <!-- Fifth Microbiological test -->
      <LabReportTestRequest testCode="VCHOL" contractedLaboratory=""
testMethod="Vibrio Cholera" analystName="Brian Brown">
       <AnalystComments>Presumptive positive detected.</AnalystComments>
       <!-- A single result test will have only one result. -->
       <ResultRequest labSampleId="N02/028681" resultName="VCHOL"
units="" resultsOfAnalysis="Detected"/>
      </LabReportTestRequest>
      <!--
      Multi-result test

       Chemical tests are an example of multi-result tests. Pesticides is
a common test which takes this structure.
        -->
      <LabReportTestRequest testCode="PESTICID" contractedLaboratory=""
testMethod="PESTICIDES" analystName="Brian Brown">
       <AnalystComments>This is a multi-result test.</AnalystComments>
       <AnalystComments/>0
       <ResultRequest labSampleId="N01/987654" resultName="Azinphos-
methyl" units="mg/kg" resultsOfAnalysis="&lt;0.20"/>
       <ResultRequest labSampleId="N01/987654" resultName="Carbaryl"
units="mg/kg" resultsOfAnalysis="&lt;0.10"/>
       <ResultRequest labSampleId="N01/987654" resultName="Chlorpyrifos"
units="mg/kg" resultsOfAnalysis="&lt;0.01"/>
       <ResultRequest labSampleId="N01/987654"
resultName="Chlorfenvinphos (cis &amp; trans)" units="mg/kg"
resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Diazinon"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Dichlorvos"
units="mg/kg" resultsOfAnalysis="&lt;0.10"/>
       <ResultRequest labSampleId="N01/987654" resultName="Dimenthoate"
units="mg/kg" resultsOfAnalysis="&lt;0.15"/>
       <ResultRequest labSampleId="N01/987654" resultName="Disulfoton"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Endosulfan (a
b &amp; sulfate)" units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Ethoprofos"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Fenamifos"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Fenitrothion"
units="mg/kg" resultsOfAnalysis="&lt;0.01"/>
       <ResultRequest labSampleId="N01/987654" resultName="Fenthion"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Fipronil"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Malathion"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Methidathion"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Mevinphos"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Monocrotophos"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Omethoate"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Parathion-
ethyl" units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Parathion-
methyl" units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Phorate"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Phosmet"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
```

```
      <ResultRequest labSampleId="N01/987654" resultName="Pirimicarb"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
      <ResultRequest labSampleId="N01/987654" resultName="Pirimiphos-
methyl" units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
     </LabReportTestRequest>
    </SampleRegistrationRecord>
   </LabReportRequest>
  </Letter>
 </BusinessContent>
</Envelope>
```

f) Example 6 – Acknowledgment

This is an example of the department acknowledging the receipt of a Sample Result Request Message sent from a Laboratory System.

```
<!--
 This is an example of the the department Certified Mail System (CMS)
acknowledging the receipt
 of an envelope containing the F0005 Sample Registration Result Letter on
behalf of eResults.
 -->
<!--
 The Acknowledgement is standard for all eResults messages and
acknowledges the receipt of an Envelope.
 It contains an Address Label that is structurally identical to the
Envelope Address Label.
-->
<Acknowledgement>
 <AddressLabel createdTimestamp="2003-01-30T09:15:00">
  <MessageIdentification id="102" typeId="R0002" typeVersion="1.0"/>
  <Sender id="eResults"/>
  <Recipient id="BERS"/>
 </AddressLabel>
 <!--
  The Response indicates whether the envelope is acknowledged (Pass) or
not acknowledged (Fail).
  In this example the envelope is acknowledged. -->
 <Response requestMessageId="12" requestMessageReceiptTime="2003-01-
30T09:10:00" outcome="Pass"/>
</Acknowledgement>
```

g) Example 7 – Sample Result Response (F0006)

This is an example of the department sending a Sample Result Response Message (in response to a Sample Result Request Message) to a Laboratory System.

```
<!--
This is an example of the eResults System responding positively to the
Sample Result Request message. It verifies that the Business Letter
content is well formed XML and passes all business rules.
-->
<!--
This example Envelope shows that a F0006 Sample Result Response Letter
is being sent from the the department to the BERS = Bernard Heath
Laboratory System .
-->
<Envelope>
 <AddressLabel createdTimestamp="2003-01-30T09:20:00">
  <MessageIdentification id="103" typeId="F0006" typeVersion="1.0"/>
  <Sender id="eResults"/>
  <Recipient id="BERS"/>
 </AddressLabel>
 <BusinessContent>
  <Letter typeId="F0006">
   <!-- This is the actual Business Letter passed to the CMS for
mailing. In this
     example it is the Sample Result Response Message (F0006).-->
   <!-- Includes the original sample result information for Report Id
RN313361. -->
   <LabReportRequest labReportId="RN313361" reportDate="2002-11-25">
    <SampleRegistrationRecord labRefId="058239"
sampleBagBarCodeReceived="IF7654321" labReceiptDate="2003-01-29"
sampleArrivalCondition="TA" tamperingEvidentCode="False">
     <ClientSample jobId="3B12349876" sampleId="1a"/>
     <!--
      Each Sample can have multiple tests requested against it.
      This example shows multiple Tests for the Sample.
      -->
     <!-- Single result test -->
     <LabReportTestRequest testCode="AFLAT" contractedLaboratory="NMI"
testMethod="AFLATOX" analystName="John Smith">
      <AnalystComments/>
      <!-- A single result test will have only one result. -->
      <ResultRequest labSampleId="N02/028681" resultName="AFLAT"
units="µg/kg" resultsOfAnalysis="&lt;5"/>
     </LabReportTestRequest>
     <!--
      Merged-result tests

      Following five tests, ECOLI, SPC, Salmonella, Staph Enterotoxin
and Vibrio Cholera are usually performed together,
      constituting the standard the department microbiologicals set of
tests
      -->
     <!-- First Microbiological test -->
     <LabReportTestRequest testCode="ECOLI" contractedLaboratory=""
testMethod="ECOLI" analystName="Brian Brown">
      <AnalystComments/>
      <!-- Each Test can have multiple results.  This example shows FIVE
RESULTS for the ECOLI Test. -->
      <ResultRequest labSampleId="N02/028681" resultName="ECOLI1"
units="cfu/g" resultsOfAnalysis="&lt;2.3"/>
      <ResultRequest labSampleId="N02/028681" resultName="ECOLI2"
units="cfu/g" resultsOfAnalysis="&lt;2.3"/>
```

```
       <ResultRequest labSampleId="N02/028681" resultName="ECOLI3"
units="cfu/g" resultsOfAnalysis="&lt;189.0"/>
       <ResultRequest labSampleId="N02/028681" resultName="ECOLI4"
units="cfu/g" resultsOfAnalysis="&lt;6.0"/>
       <ResultRequest labSampleId="N02/028681" resultName="ECOLI5"
units="cfu/g" resultsOfAnalysis="&lt;2.3"/>
      </LabReportTestRequest>
      <!-- Second Microbiological test -->
      <LabReportTestRequest testCode="SPC" contractedLaboratory=""
testMethod="SPC" analystName="Brian Brown">
       <AnalystComments/>
       <!-- Each Test can have multiple results.  This example shows FIVE
RESULTS for the SPC Test. -->
       <ResultRequest labSampleId="N02/028681" resultName="SPC1"
units="cfu/g" resultsOfAnalysis="10.4E3"/>
       <ResultRequest labSampleId="N02/028681" resultName="SPC2"
units="cfu/g" resultsOfAnalysis="2.6E4"/>
       <ResultRequest labSampleId="N02/028681" resultName="SPC3"
units="cfu/g" resultsOfAnalysis="2.4E4"/>
       <ResultRequest labSampleId="N02/028681" resultName="SPC4"
units="cfu/g" resultsOfAnalysis="1.7E3"/>
       <ResultRequest labSampleId="N02/028681" resultName="SPC5"
units="cfu/g" resultsOfAnalysis="3.5E5"/>
      </LabReportTestRequest>
      <!-- Third Microbiological test -->
      <LabReportTestRequest testCode="SALM" contractedLaboratory=""
testMethod="SALMONELLA" analystName="Brian      Brown">
       <AnalystComments/>
       <!-- Each Test can have multiple results.  This example shows FIVE
RESULTS for the SALMONELLA Test. -->
       <ResultRequest labSampleId="N02/028681"
subSampleLabReportId="RN313362" resultName="SALM1" units="cfu/25g"
resultsOfAnalysis="1"/>
       <ResultRequest labSampleId="N02/028681"
subSampleLabReportId="RN313363" resultName="SALM2" units="cfu/25g"
resultsOfAnalysis="1"/>
       <ResultRequest labSampleId="N02/028681"
subSampleLabReportId="RN313364" resultName="SALM3" units="cfu/25g"
resultsOfAnalysis="1"/>
       <ResultRequest labSampleId="N02/028681"
subSampleLabReportId="RN313365" resultName="SALM4" units="cfu/25g"
resultsOfAnalysis="2"/>
       <ResultRequest labSampleId="N02/028681"
subSampleLabReportId="RN313366" resultName="SALM5" units="cfu/25g"
resultsOfAnalysis="1"/>
      </LabReportTestRequest>
      <!-- Fourth Microbiological test -->
      <LabReportTestRequest testCode="STAPE" contractedLaboratory=""
testMethod="Staph Enterotoxin" analystName="Brian Brown">
       <AnalystComments/>
       <!-- Each Test can have multiple results.  This example shows FIVE
RESULTS for the Staph Enterotoxin Test. -->
       <ResultRequest labSampleId="N02/028681" resultName="STAPE1"
units="" resultsOfAnalysis="ND"/>
       <ResultRequest labSampleId="N02/028681" resultName="STAPE2"
units="" resultsOfAnalysis="ND"/>,
       <ResultRequest labSampleId="N02/028681" resultName="STAPE3"
units="" resultsOfAnalysis="ND"/>
       <ResultRequest labSampleId="N02/028681" resultName="STAPE4"
units="" resultsOfAnalysis="ND"/>
       <ResultRequest labSampleId="N02/028681" resultName="STAPE5"
units="" resultsOfAnalysis="ND"/>
      </LabReportTestRequest>
      <!-- Fifth Microbiological test -->
      <LabReportTestRequest testCode="VCHOL" contractedLaboratory=""
testMethod="Vibrio Cholera" analystName="Brian Brown">
       <AnalystComments>Presumptive positive detected.</AnalystComments>
       <!-- A single result test will have only one result. -->
       <ResultRequest labSampleId="N02/028681" resultName="VCHOL"
units="" resultsOfAnalysis="Detected"/>
      </LabReportTestRequest>
      <!--
      Multi-result test

       Chemical tests are an example of multi-result tests. Pesticides is
a common test which takes this structure.
       -->
      <LabReportTestRequest testCode="PESTICID" contractedLaboratory=""
testMethod="PESTICIDES" analystName="Brian Brown">
       <AnalystComments>This is a multi-result test.</AnalystComments>
       <AnalystComments/>0
       <ResultRequest labSampleId="N01/987654" resultName="Azinphos-
methyl" units="mg/kg" resultsOfAnalysis="&lt;0.20"/>
       <ResultRequest labSampleId="N01/987654" resultName="Carbaryl"
units="mg/kg" resultsOfAnalysis="&lt;0.10"/>
       <ResultRequest labSampleId="N01/987654" resultName="Chlorpyrifos"
units="mg/kg" resultsOfAnalysis="&lt;0.01"/>
       <ResultRequest labSampleId="N01/987654"
resultName="Chlorfenvinphos (cis &amp; trans)" units="mg/kg"
resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Diazinon"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Dichlorvos"
units="mg/kg" resultsOfAnalysis="&lt;0.10"/>
       <ResultRequest labSampleId="N01/987654" resultName="Dimenthoate"
units="mg/kg" resultsOfAnalysis="&lt;0.15"/>
       <ResultRequest labSampleId="N01/987654" resultName="Disulfoton"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Endosulfan (a
b &amp; sulfate)" units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Ethoprofos"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Fenamifos"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Fenitrothion"
units="mg/kg" resultsOfAnalysis="&lt;0.01"/>
       <ResultRequest labSampleId="N01/987654" resultName="Fenthion"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Fipronil"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Malathion"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Methidathion"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Mevinphos"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Monocrotophos"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Omethoate"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Parathion-
ethyl" units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Parathion-
methyl" units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Phorate"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
       <ResultRequest labSampleId="N01/987654" resultName="Phosmet"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
```

```
      <ResultRequest labSampleId="N01/987654" resultName="Pirimicarb"
units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
      <ResultRequest labSampleId="N01/987654" resultName="Pirimiphos-
methyl" units="mg/kg" resultsOfAnalysis="&lt;0.05"/>
     </LabReportTestRequest>
    </SampleRegistrationRecord>
   </LabReportRequest>
   <Response requestMessageId="12" requestMessageReceiptTime="2003-01-
30T09:16:00" outcome="Pass"/>
  </Letter>
 </BusinessContent>
</Envelope>
```

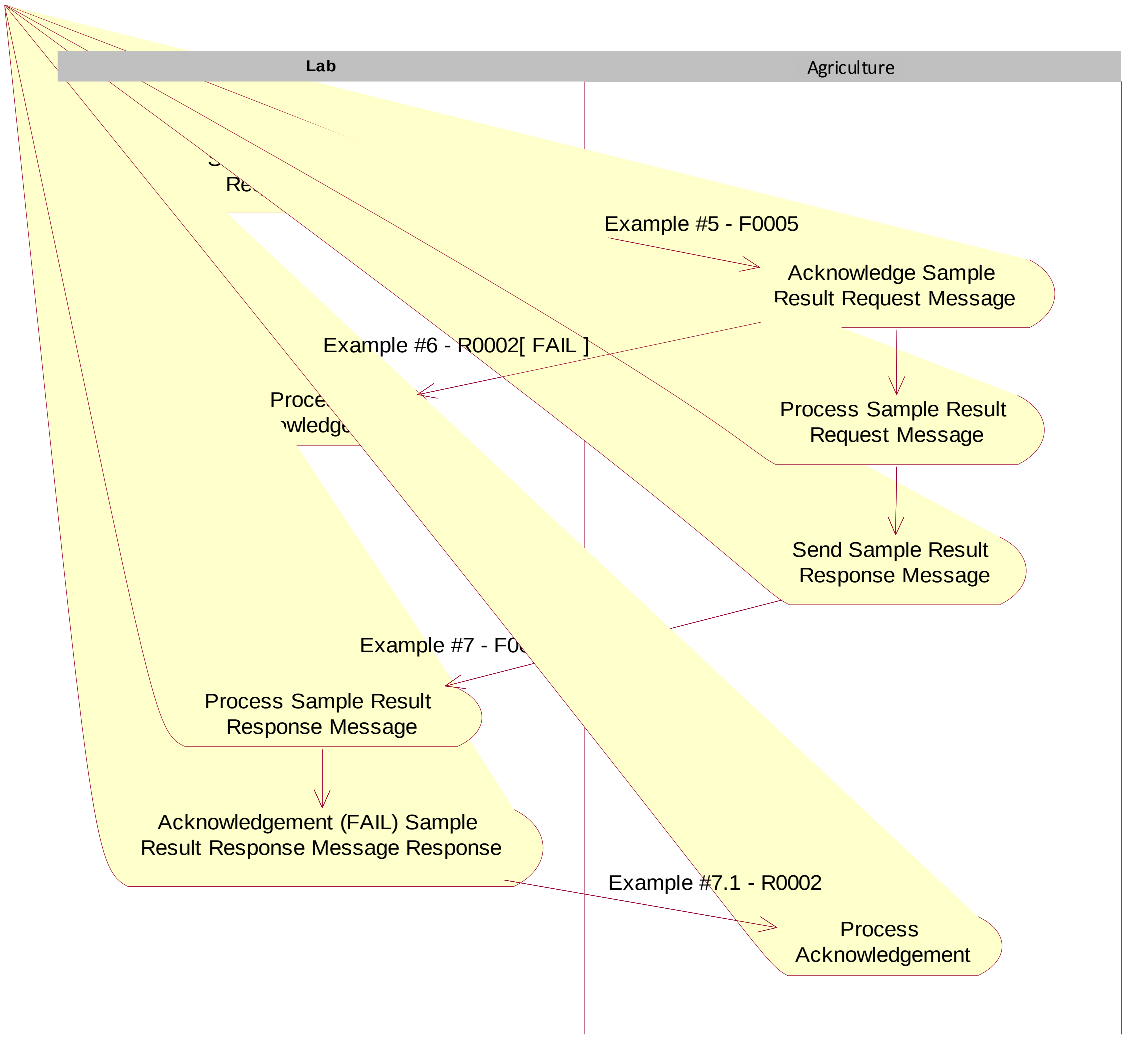
h) Example 8 – Acknowledgment (PASS)

This is an example of a Laboratory System acknowledging the receipt of a Sample Result Response Message sent from the department.

```
<!--
This is an example of the Laboratory Certified Mail System (CMS)
acknowledging the receipt of the envelope containing the F0006 Sample
Result Response Letter sent by the department.
-->
<!--
The Acknowledgement is standard for all eResults messages and
acknowledges
the receipt of an Envelope.  It contains an Address Label that is
structurally
identical to the Envelope Address Label.
-->
<Acknowledgement>
 <AddressLabel createdTimestamp="2003-02-01T09:15:00">
  <MessageIdentification id="12" typeId="R0002" typeVersion="1.0"/>
  <Sender id="BERS"/>
  <Recipient id="eResults"/>
 </AddressLabel>
 <!--
   The Response indicates whether the envelope is acknowledged (Pass) or
not acknowledged (Fail).
    In this example the envelope is acknowledged.
  -->
 <Response requestMessageId="103" requestMessageReceiptTime="2003-01-
29T09:14:50" outcome="Pass"/>
</Acknowledgement>
```

Unsuccessful Acknowledgment

This example describes the unsuccessful end-to-end Sample Result delivery, acknowledgement and response process.


Lab
Agriculture
Re
Example #5 - F0005
Acknowledge Sample Result Request Message
Example #6 - R0002[FAIL]
Proce
wledge
Process Sample Result Request Message
Send Sample Result Response Message
Example #7 - F0
Process Sample Result Response Message
Acknowledgement (FAIL) Sample Result Response Message Response
Example #7.1 - R0002
Process Acknowledgement

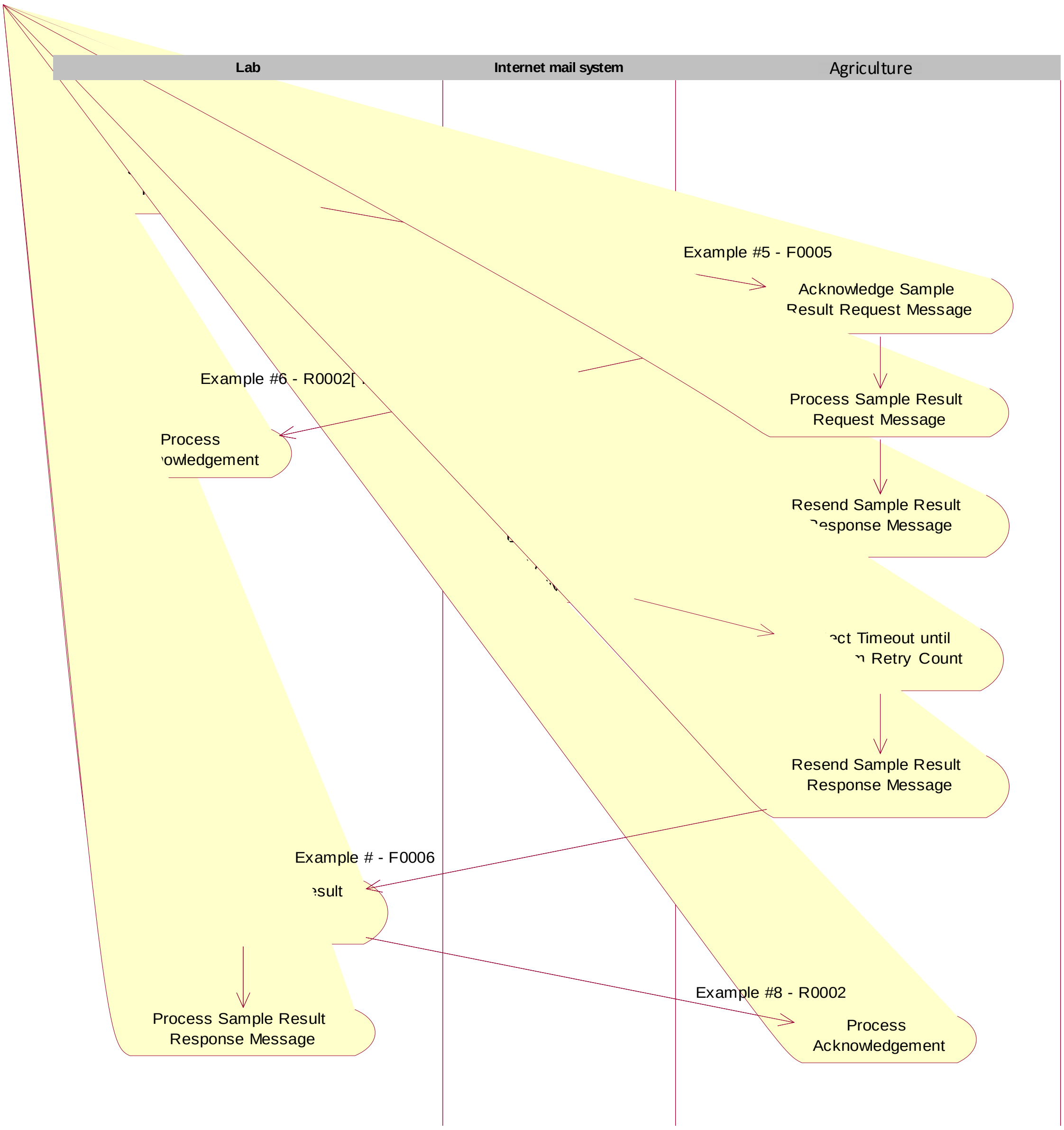
6.3.1 Example 7.1 – Negative Acknowledgement (FAIL)

This is an example of a Laboratory System not acknowledging the receipt of a Sample Result Response Message sent from the department.

```
<!--
 This is an example of the Laboratory Certified Mail System (CMS)
 not acknowledging the receipt of the envelope containing the F0004
Sample
 Registration Response Letter sent by the department because of a CRC
failure.
 -->
<!--
 The Acknowledgement is standard for all eResults messages and
acknowledges the receipt of an Envelope.
 It contains an Address Label that is structurally identical to the
Envelope Address Label.
-->
<Acknowledgement>
 <AddressLabel createdTimestamp="2003-01-29T09:15:00">
  <MessageIdentification id="11" typeId="R0002" typeVersion="1.0"/>
  <Sender id="BERS"/>
  <Recipient id="eResults"/>
 </AddressLabel>
 <!--
  The Response indicates whether the envelope is acknowledged (Pass) or
not acknowledged (Fail).
  In this example the envelope is not acknowledged.
  -->
 <Response requestMessageId="103" requestMessageReceiptTime="2003-01-
29T09:14:50" outcome="Fail">
  <!-- An error code and description is provided for the envelope that
is not acknowledged. -->
  <Error errorCode="E0038">
   <Detail>Invalid Message Checksum</Detail>
  </Error>
 </Response>
</Acknowledgement>
```

Timeout Error in Sample Result delivery process

This example describes the unsuccessful end-to-end Sample Result delivery, acknowledgement and response process due a to timeout error. Refer to the XML references defined in earlier examples.


Lab
Internet mail system
Agriculture
Example #5 - F0005
Acknowledge Sample Result Request Message
Process Sample Result Request Message
Example #6 - R0002
Process owledgement
Resend Sample Result esponse Message
ect Timeout until n Retry Count
Resend Sample Result Response Message
Example # - F0006
esult
Process Sample Result Response Message
Example #8 - R0002
Process Acknowledgement

Appendix D – XML Schema Definitions

This section will provide the schema definitions for the eResults Messaging Service.

XML Schema Hierarchy

The eResults XML schema defines two types of XML documents – Envelopes and Acknowledgments.

a) XML Structure for an Envelope Document

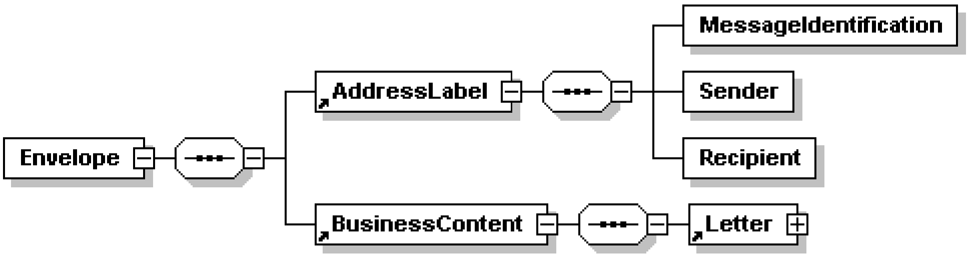


Figure 16 – Envelope Structure

b) XML Structure for an Acknowledgment Document

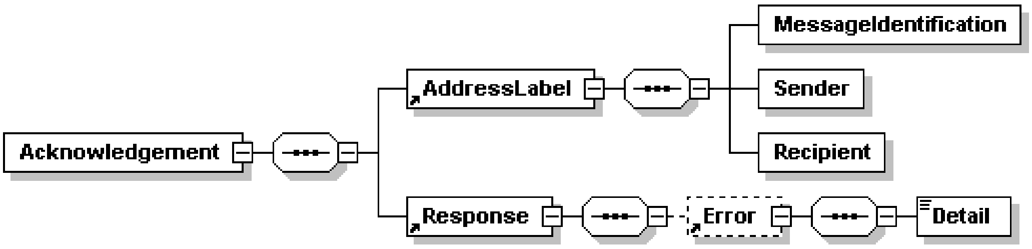


Figure 17 - Acknowledgment Structure

c) XML Structure for a Letter within an Envelope Document

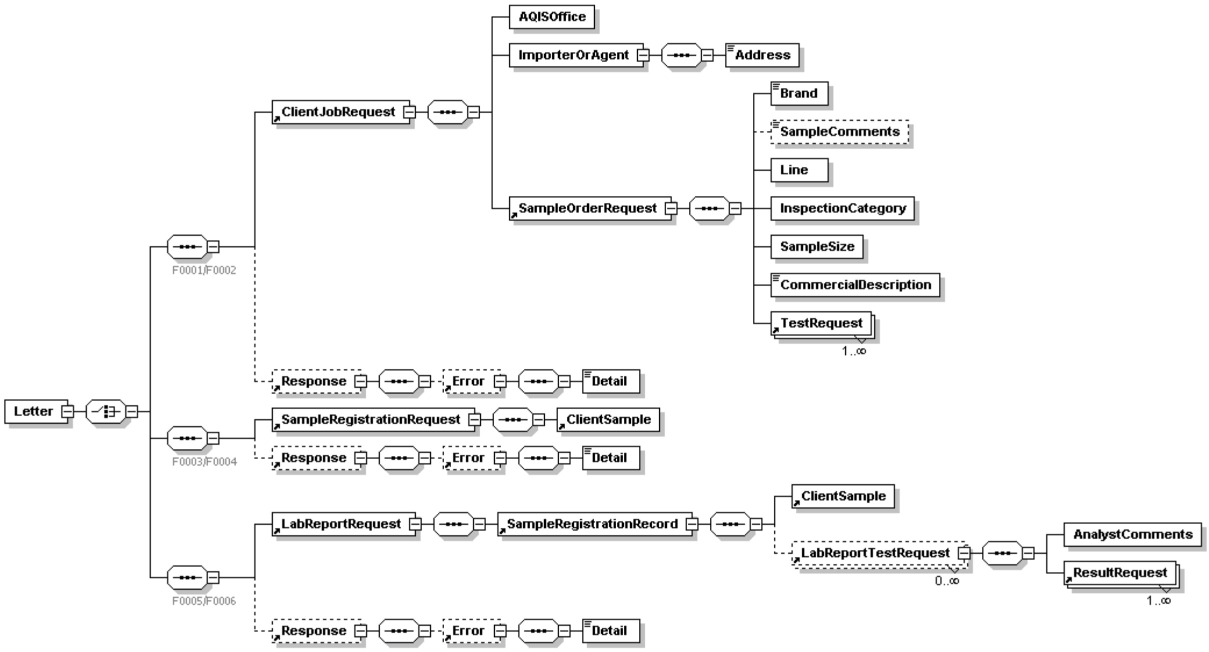


Figure 18 – Letter Structure

eResults Messaging Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 3 U (http://www.xmlspy.com) by Damana
Madden (Company) -->
<!--W3C Schema generated by XMLSPY v5 rel. 3 U (http://www.xmlspy.com)-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" id="eResultMessaging">
 <xs:element name="Envelope">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="AddressLabel"/>
    <xs:element ref="BusinessContent"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="Acknowledgement">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="AddressLabel"/>
    <xs:element ref="Response"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:element name="AddressLabel">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="MessageIdentification">
     <xs:complexType>
      <xs:attribute name="id" type="xs:positiveInteger" use="required"/>
      <xs:attribute name="typeId" use="required">
       <xs:simpleType>
        <xs:restriction base="xs:string">
         <xs:length value="5"/>
        </xs:restriction>
       </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="typeVersion" use="required">
       <xs:simpleType>
        <xs:restriction base="xs:string">
         <xs:minLength value="3"/>
         <xs:maxLength value="5"/>
        </xs:restriction>
       </xs:simpleType>
      </xs:attribute>
     </xs:complexType>
    </xs:element>
    <xs:element name="Sender">
     <xs:complexType>
      <xs:attribute name="id" use="required">
       <xs:simpleType>
        <xs:restriction base="xs:string">
         <xs:minLength value="1"/>
         <xs:maxLength value="8"/>
        </xs:restriction>
       </xs:simpleType>
      </xs:attribute>
     </xs:complexType>
    </xs:element>
    <xs:element name="Recipient">
     <xs:complexType>
      <xs:attribute name="id" use="required">
       <xs:simpleType>
        <xs:restriction base="xs:string">
         <xs:minLength value="1"/>
         <xs:maxLength value="8"/>
        </xs:restriction>
       </xs:simpleType>
      </xs:attribute>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
   <xs:attribute name="createdTimestamp" use="required">
    <xs:simpleType>
     <xs:restriction base="xs:dateTime"/>
    </xs:simpleType>
   </xs:attribute>
  </xs:complexType>
 </xs:element>
 <xs:element name="Error">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Detail">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="255"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:element>
   </xs:sequence>
   <xs:attribute name="errorCode" use="required">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:length value="5"/>
      <xs:pattern value="E\d{4}"/>
     </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:element name="BusinessContent"/>
  <xs:element name="Response">
   <xs:complexType>
    <xs:sequence>
     <xs:element ref="Error" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="requestMessageId" type="xs:positiveInteger"
 use="required"/>
    <xs:attribute name="requestMessageReceiptTime" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:dateTime"/>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="outcome" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="Fail"/>
       <xs:enumeration value="Pass"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:element name="LabReportRequest">
   <xs:complexType>
    <xs:sequence>
     <xs:element ref="SampleRegistrationRecord"/>
    </xs:sequence>
    <xs:attribute name="labReportId" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="15"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="reportDate" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:date"/>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:element name="LabReportTestRequest">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="AnalystComments"/>
     <xs:element ref="ResultRequest" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="testCode" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="8"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="contractedLaboratory" use="optional">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="0"/>
       <xs:maxLength value="70"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="testMethod" use="optional">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="0"/>
       <xs:maxLength value="30"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="analystName" use="optional">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="0"/>
       <xs:maxLength value="50"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:element name="ResultRequest">
   <xs:complexType>
    <xs:attribute name="labSampleId" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="30"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="subSampleLabReportId" use="optional">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="0"/>
       <xs:maxLength value="50"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="resultName" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="70"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="units" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="0"/>
       <xs:maxLength value="20"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="resultsOfAnalysis" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="50"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:element name="SampleRegistrationRequest">
   <xs:complexType>
    <xs:sequence>
     <xs:element ref="ClientSample"/>
    </xs:sequence>
    <xs:attribute name="labRefId" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="50"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="sampleBagBarCodeReceived" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:pattern value="IF\d{7}"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="labReceiptDate" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:date"/>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="sampleArrivalCondition" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="5"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="tamperingEvidentCode" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="True"/>
       <xs:enumeration value="False"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:element name="SampleRegistrationRecord">
   <xs:complexType>
    <xs:sequence>
     <xs:element ref="ClientSample"/>
     <xs:element ref="LabReportTestRequest" minOccurs="0"
 maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="labRefId" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="50"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="sampleBagBarCodeReceived" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:pattern value="IF\d{7}"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="labReceiptDate" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:date"/>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="sampleArrivalCondition" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="5"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="tamperingEvidentCode" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="True"/>
       <xs:enumeration value="False"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:element name="Letter">
   <xs:complexType>
    <xs:choice>
     <xs:sequence>
      <xs:annotation>
       <xs:documentation>F0001/F0002</xs:documentation>
      </xs:annotation>
      <xs:element ref="ClientJobRequest"/>
      <xs:element ref="Response" minOccurs="0"/>
     </xs:sequence>
     <xs:sequence>
      <xs:annotation>
       <xs:documentation>F0003/F0004</xs:documentation>
      </xs:annotation>
      <xs:element ref="SampleRegistrationRequest"/>
      <xs:element ref="Response" minOccurs="0"/>
     </xs:sequence>
     <xs:sequence>
      <xs:annotation>
       <xs:documentation>F0005/F0006</xs:documentation>
      </xs:annotation>
      <xs:element ref="LabReportRequest"/>
      <xs:element ref="Response" minOccurs="0"/>
     </xs:sequence>
    </xs:choice>
    <xs:attribute name="typeId" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="F0001"/>
       <xs:enumeration value="F0002"/>
       <xs:enumeration value="F0003"/>
       <xs:enumeration value="F0004"/>
       <xs:enumeration value="F0005"/>
       <xs:enumeration value="F0006"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
```

```
</xs:element>
<xs:element name="ClientSample">
 <xs:complexType>
  <xs:attribute name="jobId" use="required">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:minLength value="1"/>
     <xs:maxLength value="10"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="sampleId" use="required">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:minLength value="1"/>
     <xs:maxLength value="30"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
 </xs:complexType>
</xs:element>
<xs:element name="SampleOrderRequest">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Brand">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="50"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:element>
   <xs:element name="SampleComments" minOccurs="0">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="255"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:element>
   <xs:element name="Line">
    <xs:complexType>
     <xs:attribute name="number" type="xs:integer" use="required"/>
     <xs:attribute name="countryOfOrigin" use="required">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="2"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
    </xs:complexType>
   </xs:element>
   <xs:element name="InspectionCategory">
    <xs:complexType>
     <xs:attribute name="referredCode" use="required">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="2"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
     <xs:attribute name="actualCode" use="required">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="2"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
    </xs:complexType>
   </xs:element>
   <xs:element name="SampleSize">
    <xs:complexType>
     <xs:attribute name="count" type="xs:integer" use="required"/>
     <xs:attribute name="quantity" type="xs:integer" use="required"/>
     <xs:attribute name="unitsCode" use="required">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="3"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
    </xs:complexType>
   </xs:element>
   <xs:element name="CommercialDescription">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="255"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:element>
   <xs:element ref="TestRequest" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="sampleId" use="required">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:minLength value="1"/>
     <xs:maxLength value="30"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="lotCode" use="optional">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:minLength value="1"/>
     <xs:maxLength value="30"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="sampleBagBarcode" use="required">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:pattern value="IF\d{7}"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="sampleDate" type="xs:date" use="required"/>
  <xs:attribute name="sampleTime" type="xs:time" use="required"/>
  <xs:attribute name="sampleInspectionConditionCode" use="required">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:minLength value="1"/>
     <xs:maxLength value="5"/>
    </xs:restriction>
```

```
   </xs:simpleType>
  </xs:attribute>
 </xs:complexType>
</xs:element>
<xs:element name="ClientJobRequest">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="AQISOffice">
    <xs:complexType>
     <xs:attribute name="name" use="required">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="30"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
     <xs:attribute name="faxNumber" use="required">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="20"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
     <xs:attribute name="authorisedOfficer" use="required">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="10"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
    </xs:complexType>
   </xs:element>
   <xs:element name="ImporterOrAgent">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="Address">
       <xs:simpleType>
        <xs:restriction base="xs:string">
         <xs:minLength value="1"/>
         <xs:maxLength value="140"/>
        </xs:restriction>
       </xs:simpleType>
      </xs:element>
     </xs:sequence>
     <xs:attribute name="name" use="required">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="50"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
     <xs:attribute name="phoneNumber" use="optional">
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="20"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:attribute>
     <xs:attribute name="faxNumber" use="optional">
```

```
       <xs:simpleType>
        <xs:restriction base="xs:string">
         <xs:minLength value="1"/>
         <xs:maxLength value="20"/>
        </xs:restriction>
       </xs:simpleType>
      </xs:attribute>
     </xs:complexType>
    </xs:element>
    <xs:element ref="SampleOrderRequest"/>
   </xs:sequence>
   <xs:attribute name="jobId" use="required">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:minLength value="9"/>
      <xs:maxLength value="10"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:attribute>
  </xs:complexType>
 </xs:element>
 <xs:element name="TestRequest">
  <xs:complexType>
   <xs:attribute name="testCode" use="required">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="8"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:attribute>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

End of Document.