Mail.XML 11.0D Technical Specification

Data Distribution

Version 10.0

Release 29 January 2012

Release 30 April 2012

Release 31 June 2012

Release 32 October 2012

Release 33 January 2013

Release 34 April 2013

February 8, 2013

Prepared by

United States Postal Service

Technical Support from the USPS Help Desk and Tier 2 Support:

The USPS provides Mail.XML technical support for mailers/vendors and support is only specific to the business rules it pertains to implementation of business transactions, error codes/descriptions and exception handling with the Mail.XML implementations.

The USPS does not provide nor it is equipped to provide technical support for software development tools or IDEs (Integrated Development Environments) like Visual Studio, and software development related activities as they pertain to different technology specific programming environments for different companies, such as .Net, Java, Visual Basic, Power Builder, C, C++ etc.

The Mailer/vendor is responsible for obtaining technical support and assistance directly from the IDE vendor such as Microsoft or IBM or Sun/Oracle for software development related

TABLE OF CONTENTS

DOCUMENT CHANGE HISTORY

These are the changes from Mail.XML 11.0D Data Distribution Version 9.0 to Mail.XML 11.0D Data Distribution Version 10.0

Date	Section #	Title	Description
1/18/13			Updated footer and header of the document
1/18/13			Assigned a new style to the document

These are the changes from Mail.XML 11.0D Data Distribution Version 8.0 to Mail.XML 11.0D Data Distribution Version 9.0

Date	Section #	Title	Description
11/27/12	2.1.2	How to implement available Mail.XML versions for Full-Service Data Feedback?	Removed "/use" from the section heading to eliminate the redundancy
11/27/12	2.1.4	What is Push Model/Subscription method?	Under "Getting Ready for Push Data Distribution Checklist": •Removed extra space between "are" and "outlined" in the first sentence. •Removed the phrase "Refer to the" just before the hyperlink for the TEM Checklist
11/27/12	1.7	Purpose	Revised last sentence
11/27/12	2.1.1	What is needed to receive data through the data exchange methods identified?	Added "be" between "must" and "registered" in the first sentence under "Step1"
11/27/12	2.1.7	Error/ Return Codes	In the last sentence of first paragraph: •Added "all" before "versions" •Added a period at the end of the sentence
11/27/12	All		Edit the document to removed misleading prefix "FullService" in messages and data blocks
11/27/12	7.0	Appendix C	Removed references to Mail.XML 12.0A in the text under "MESSAGE-TYPE PUSH WSDL"

These are the changes from Mail.XML 11.0D Data Distribution Version 7.0 to Mail.XML 11.0D Data Distribution Version 8.0

Date	Section #	Title	Description

Date	Section #	Title	Description
11/05/12			Updated the front page to reflect the PostalOne! release numbers instead of that of Full Service i.e January 2013 release number becomes 33 (PO!) instead of 12 (FS)
10/31/12	4.8	FullServiceAddressCorrection Delivery	Corrected the spelling of the attribute from "LargeTransactionDivideroptionalGroupType" to "LargeTransactionDividerGroupOptionalType"
10/31/12	1.5	Roadmap of Mail.XML	Updated the roadmap of supported version till April 2013
10/31/12	All		Removed references to generic Mail.XML version such as 9.0X or 9.X

These are the changes from Mail.XML 11.0D Data Distribution Version 6.0 to Mail.XML 11.0D Data Distribution Version 7.0

Date	Section #	Title	Description
9/27/12	All		Updated front page and footer for September 28 publication

These are the changes from Mail.XML 11.0D Data Distribution Version 5.0 to Mail.XML 11.0D Data Distribution Version 6.0

Date	Section #	Title	Description
9/04/12	6.0	Appendix B	Updated table formatting to be consistent and made headers repeat across multiple pages.
9/04/12	2.2-2.4		Edited the line spacing between paragraphs
9/04/12	3.0	Message Choreography for Data Distribution	Removed extra page breaks between sections 3 and 4
9/04/12	4.1	ByForConflictNotification	Updated dates for releases in the note
9/04/12	4.16.1	DataQualityVerificationRepor tDelivery	Removed yellow highlights from elements in the table
9/04/12	5.0	Appendix A	Added the opening and closing indicators such "complexTypeName BEGINS" and "compleTypeName ENDS" where missing to tables in Appendix A.
9/04/12	5.0	Appendix A	Adding
9/04/12	5.0	Appendix A	Removed extra page break between sections 5 and 6
9/04/12	7.0	Appendix C	Removed extra page break between Appendix

Date	Section #	Title	Description
			C and Index

These are the changes from Mail.XML 11.0D Data Distribution Version 4.0 to Mail.XML 11.0D Data Distribution Version 5.0

Date	Section #	Title	Description
08/16/12			Updated cover page, header and footer
08/16/12	1.15	Exceptions	Updated table for current version, release, & date information
08/16/12	1.6	Roadmap of Mail.XML	Updated text to reflect current status and versions Added column in table 1 for January 2013 – Release 33.0

These are the changes from Mail.XML 11.0D Data Distribution Version 3.2 to Mail.XML 11.0D Data Distribution Version 4.0

Date	Section #	Title	Description
07/03/12	All		Fixed formatting and grammatical errors throughout the document

These are the changes from Mail.XML 11.0D Data Distribution Version 3.1 to Mail.XML 11.0D Data Distribution Version 3.2

Date	Section #	Title	Description
05/15/12	All		Removed watermark and updated the document cover and footer for May 18 Initial Publication for RIBBS for October 2012 Release
05/15/12	2.2	Profile Management Overview	Removed this section because it has been extracted from the Data Distribution and documented in the MID-CRID technical specification
05/15/12	6.0	Appendix B – Mail.XML 11.0D Simple Types Data Structure	Removed “NOTE#1: Mail.XML 9.0B and 9.0C versions will no longer be supported from January2012 thus all of the references of 9.0B and 9.0C have been removed or limited to the minimum possible from this guide including the simple

Date	Section #	Title	Description
			types definitions." because the note has become irrelevant
05/11/12	6.0	Appendix B – Mail.XML 11.0D Simple Types Data Structure	Deleted the below simple types from the Technical specification because they are no longer supported in the Mail.XML 11.0D XSD bundleLevelType componentWeightType container24CharacterBarcodeType containerGroupingDescriptionType cptPeriodicalAdPercentageType cptPeriodicalAdPercentageBasisType cptPeriodicalAdPercentageStatusType daysType dec03-2 dec04-3 dec18-6 dec10-4 eMD8125ASNBarcodeType es2 es31 fASTContentIDType incentiveTypeType inOutCountyIndicatorType int04 maildatRecordStatus mailIssueType mailXMLStatusType moveUpdateType ns08 ns13 n04 newAddressType

Date	Section #	Title	Description
			periodicalStatementType postageStatementStateType pvdsContainerSubType recipientType s31 s72 s255 segmentIDType unloadStateType USPSPriceIncentiveType

These are the changes from Mail.XML 11.0D Data Distribution Version 3.0 to Mail.XML 11.0D Data Distribution Version 3.1

Date	Section #	Title	Description
05/07/12	All		Updated cover and version numbering for May 18 Final Publication for RIBBS
04/25/12	All		Updated the footer , header and front-page
04/25/12	All		Added DRAFT watermark seal

These are the changes from Mail.XML 11.0D Data Distribution Version 2.1 to Mail.XML 11.0D Data Distribution Version 3.0

Date	Section #	Title	Description
04/09/12	All		Updated cover and version numbering for April 10 Final Publication for RIBBS
04/09/12	All		Removed the DRAFT watermark seal

These are the changes from Mail.XML 11.0D Data Distribution Version 2.0 to Mail.XML 11.0D Data Distribution Version 2.1

Date	Section #	Title	Description
03/13/12	All		Updated document front page and footer
03/13/12	1.5	USPS® Limitation Rule on Transaction Requests	Added a disclaimer to inform users/mailers the statement and limitation in this section are under review by the USPS. Updates for this section will be provided in the next Mail.XML technical specification publication.

These are the changes Version 1.2.3 to Version 2.0

Date	Section #	Title	Description
03/05/2012	2.1.5	What is Pull Model?	Revised the text in the second paragraph to be concise
03/05/2012	7.0	Appendix C – WSDLs and XSDs	Updated the description of Mail.XML 10.0A and 11.0D MESSAGE-TYPE PUSH WSDL
03/02/2012	All		Updated footer and header
03/02/2012			Added a new disclaimer on the second page of the document

These are the changes Version 1.2.2 to Version 1.2.3

Date	Section #	Title	Description
02/28/2012	1.6	Roadmap of Mail.XML	Removed messages with strikethrough from the table of supported messaged
02/27/2012	6.0	Appendix B – Mail.XML 11.0D Simple Types Data Structure	Fixed the formatting of subsections to match 6.X enumeration
02/24/2012	All		Updated document, revised formatting and added “DRAFT” watermark

These are the changes from Version 1.2.1 to Version 1.2.2

Date	Section #	Title	Description
01/26/2012	Appendix A 5.5		Added FullServiceIndicator before ContainerVisibilityEntry ENDS in Container Visibility Entry
01/26/2012	Appendix A 5.5		Added Choice Block BEGINS and Choice Block END wrapped around missing elements IMcb, IMtb, IMpb in Container Visibility Entry
01/26/2012	Appendix A 5.5		Added ContainerID, ParentContainerID, SiblingContainerID, ContainerType before USPSFacilityLocaleKey in Container Visibility Entry:
01/26/2012	Appendix A 5.5		Updated Container Visibility Entry: Removed Choice Block BEGINS before ConsigneeApptID and removed Choice Block END after CSAID
01/26/2012	Appendix A 5.5		Updated Container Visibility Entry: ConsigneeApptID, business rules column to say “Optional;”; either/or

Date	Section #	Title	Description
			with CSAID should not be the requirement
01/26/2012	Appendix A 5.5		Updated Container Visibility Entry: UserLicenseCode and MaildatJobID, business rules column to say “Required when Mail.dat is used”
01/26/2012	4.14		Updated third to last bullet, *note to say “*Name of USPS facility or Airport where scan occurred”
01/26/2012	4.13.1		Updated ConsigneeApptID and CSAID, business rules column to say “Optional”; either/or with CSAID should not be the requirement
01/26/2012	4.13.1		Updated CustomerGroupID, business rules column to say “Required when Mail.dat is used”
01/26/2012	4.13.1		Updated UserLicenseCode and MaildatJobID, business rules column to say “Required when Mail.dat is used”
01/26/2012	4.12		Updated second to last bullet, *note should to say “*Name of USPS facility or Airport where scan occurred”
01/26/2012	4.12		Removed last bullet which refers to ACS services in ContainerVisibilityDelivery
01/26/2012	4.11.1		Updated CustomerGroupID, business rules column to say “Optional when Mail.XML is used”
01/26/2012	4.11.1		Updated UserLicenseCode and MaildatJobID, business rules column to say “Required when Mail.dat is used”
01/26/2012	2.1		FTP paragraph below table: rewrote second sentence (beginning with “Any USPS…”) for clarity & grammar
01/26/2012	1.10		Rewrote this section for clarity & grammar
01/26/2012			Made sure all references to “*PostalOne!*” are used as an adjective and formatted correctly

Date	Section #	Title	Description
			(e.g., "the *PostalOne*! system")
01/26/2012			Added label and caption to all tables
01/26/2012			Included a page break between the change history and the start of section 1
01/26/2012			Included a page break before the Index
01/26/2012			Updated copyright date in footer to 2012

These are the changes from Version 1.2.0 to Version 1.2.1

Dates	Section #	Title	Description
01/20/2012	All		Updated document version in the footer and front page
01/20/2012	All		Removed the watermark from the document
01/20/2012	All		Removed "DRAFT" from the document title

These are the changes from Version 1.1.0 to Version 1.2.0

Dates	Section #	Title	Description
01/09/2012	All		Edited document front page to include Release 8
01/09/2012	All		Updated document version and date in the footer

These are the changes from RIBBS Version 1.0.0 to Version 1.1.0 (Dec 30, 2011).

Dates	Section #	Title	Description
12/30/2011	All		Replaced "Refer this complex type in Appendix A" by "Refer to this complex type in Appendix A"
12/30/2011	All		Replaced "Refer this simple type in Appendix B" by "Refer to this simple type in Appendix B"
12/30/11	5.5	ContainerVisibilityEntry	Specified ConsigneeApptID in ContainerVisibilityEntry to be

Dates	Section #	Title	Description
			Optional but (Required when CSADispatchNumber is not provided)
12/30/2011	4.13.1	ContainerVisibilityQueryRequest Data Structure	Updated ConsigneeApptID from a string of 9 to 12 characters in ContainerVisibilityQueryRequest message
12/30/2011	1.1	Program Overview	Updated the texts respective to the overview of each Mail.XML version
12/15/2011		Header & Footer	Updated Header & Footer

These are the changes from RIBBS Version 9.3.8.0 to Version 1.0.0 (Nov 15, 2011).

Dates	Section #	Title	Description
11/10/2011		Appendix A – Mail.XML 11.0D Complex and Attribute Groups Definitions	Updated byForConflictType from s50 to string
11/10/2011		Appendix A – Mail.XML 11.0D Complex and Attribute Groups Definitions	Updated ConsigneeApptID and CSADispatchNumber inside ContainerVisibilityQueryRequest
11/07/2011	All	Footer and Header	Updated footer and header

These are the changes from RIBBS Version 9.3.8 (Aug 9, 2011) for release 29 (Jan 8) to RIBBS Version 9.3.8 (Aug 9, 2011).

These changes are being implemented to contribute to the goal of breaking down and editing the Postal Service Mail.XML technical specifications by Mail.XML supported versions to facilitate editing and enhance readability and navigation throughout the document.

In addition, to easier navigation, some sections such as the appendixes have been reorganized.

Previously the technical specifications were just divided by function (eDoc, Data Distribution, FAST). In that scheme, all supported Mail.XML for each release were all combined in one technical specification per function release. i.e. For Release 28 (November 2011) the Mail.XML eDoc technical specification included the supported Mail.XML versions 9.0B, 9.0C and 10.0C, as while the FAST technical specification containing Mail.XML 8.0B and 10.0 versions.

This approach resulted sometimes in bulky document that became difficult to edit, manage or read as with documents mounting up to sometimes 1000 pages.

With the new editing process, the technical guides are edited focusing on the Mail.XML supported version per function.

For instance, the FAST guide supporting Mail.XML 8.0B and 10.0 versions will have two different sub-Technical specifications for Release 29 (January 2012).

Each technical specification will also group all Mail.XML version and their respective supported errata versions.

For instance, 9.0B and 9.0C supported in eDoc technical specification for Release 28 are combined in one document since they are errata for Mail.XML 9.0. However if Mail.XML 8.0B and 8.1 were supported, they would have been in two different documents since they are not of the same Mail.XML version.

By convention, Mail.XML version name containing a letter, such as 7.0C, 9.0B, 10.0A are considered errata. In contrast, Mail.XML 8.0 and 8.1 are considered two different versions.

Date	Section #	Title	Description
09/29/2011	All		Replaced Mail.XML 11.0A by Mail.XML 11.0B throughout the document to update the version supported in January 2012.
09/29/2011	4.0	Mail.XML 11.0 Data Feedback and Profile Management and Miscellaneous Messages	Removed FullSerive word from the highlighted four NonComplianceDataWithPostageOwed and NixieDetal messages to comply to reflect Mail.XML 11.0 requirements
9/07/2011	6.0	Appendix B – Mail.XML 11.0B Simple Types Data Structure	Updated the rateCateoryType and DMLType, and mail.XMLContainerType based on Mail.XML 11.0B CRs as of 9/05/2011
9/01/2011		Change History	Added a new column “Date” to the table of Change History to record the dates of implemented changes.
8/31/2011	5.0	Appendix A	Added new headers to the tables of Complex and Attribute Data Types Data Structure for Mail.XML 10.0A
8/31/2011	5.0	Appendix A	Added new headers to the tables of Complex and Attribute Data Types Data Structure for Mail.XML 11.0B
8/31/2011	4.0	Mail.XML 11.0 Data Feedback and Profile Management and Miscellaneous Messages	Added headers to the table of Mail.XML 11.0B messages
8/31/2011	4.0	Mail.XML 11.0 Data Feedback and Profile Management and Miscellaneous Messages	Added headers to the table of Mail.XML 11.0B messages
8/31/2011	5.0,6.0,7.0	Appendix A, B, C	Reorganized the appendixes order
8/24/2011		Footer	Updated the footer to include USPS® – Postal Service Mail.XML 11.0B Technical Specification Version 1.0

Date	Section #	Title	Description
8/24/2011	All		Extracted Mail.XML Technical Specification from the previous Data Distribution Technical Specification submitted on RIBBS on August 9, 2011

Postal Service Mail.XML Technical Specification

Program Overview

The *PostalOne!*® system is an electronic suite of services being developed for business mailers by the United States Postal Service® to improve the mailing process. The system takes advantage of the advances in technology and leverages them to continuously improve and simplify the mailing experience.

Customers can electronically send detailed information to *PostalOne!* system using the mailing industry's standardized Mail.XML™ format. Mail.XML is an XML-based IDEAlliance® specification for web services. Web services use simple HTTP(s) protocols to communicate data over the Internet, bypassing technology-specific restrictions and avoiding network-related security rules. See the IDEAlliance Mail.XML specifications at http://www.idealliance.org

All business functions supported by the USPS as part of the Mail.XML implementation are based on SOA (Services Oriented Architecture). This architecture enables a mailer's software to manage the data transactions to query, create, update, and cancel various data sets representing business functions and subscription model. Additionally, mailers' software is able to receive synchronous and asynchronous transactions through Web Services protocol from the USPS in support of the just-in-time communication architecture.

The Postal Service's implementation of Mail.XML complements Mail.dat. Mailers using Mail.dat can continue to use it as a database, but can use Mail.XML for communication and automation of business –to- business processing. Mail.XML allows business function-specific communication and faster change management processes with little to no impact to other service providers. For example, you can add optional fields or new messages without requiring software in the mailing supply chain to change.

Mail.XML allows the support of electronic documentation and end-to-end mailing supply chain business process management for a multitude of business processes, including qualification reports, container and bundle reports, spoilage, eInduction, containerization nesting, postage statements, postage adjustments, postage status query information, piece detail, piece ranging, and many other business process related communications. Mail.XML also provides the ability to electronically access USPS Full-Service data and to perform business functions related to Full-Service. Mail.XML supports copalletization, mixed mailing, manifest mailing, and business processes specific to MLOCR and continuous mailing environments.

One fundamental difference between the Mail.dat files and Mail.XML messages is the information content. For example, for qualification reports and postage statements Mail.dat calculates from the raw data in several Mail.dat files, the copies and pieces at each price level in the qualification report and the postage statements. This sometimes leads to round-off differences as compared to qualification reports and postage statements calculated by the mailer using the same Mail.dat file. In Mail.XML, the mailer provides the numbers of copies and pieces for each price level directly and there are less round-off errors.

The Postal Service Mail.XML Technical Specification, is documented by subject matter into the following volumes:

- Postal Service Mail.XML Technical Specification for Postage Payment & Reporting (eDoc)
- Postal Service Mail.XML Technical Specification for Appointment Scheduling (FAST)
- Postal Service Mail.XML Technical Specification for Profiles & Full-Service Feedback

This document focuses solely on the data distribution message functionalities for Full-Service data.

Mail.XML 11.0D

Mail.XML 11.0D corrects technical errors from previous Mail.XML versions in an effort to support and update the USPS automation vision for 2012. New features were added In Mail.XML 11.0D. This document lists changes for Mail.XML 11.0D version. A later version beyond 11.0D may be implemented, in which case an updated Mail.XML Data Distribution technical specification will be made available.

This technical specification does not include the IDEAlliance Mail.XML change requests (CRs) processed after October 31, 2011. The next version of this document will include those CRs.

The Data Distribution system will continue to support Mail.XML 10.0. In previous versions such as Mail.XML 8.1, the messages in XSDs are tightly coupled to one main XSD and one base XSD. Previously, the main XSD covered all message sets for the eDoc, FAST, and Data Distribution. This tight coupling required users to validate the complete XSD. Modularization was introduced in Mail.XML 9.0 and onwards, where each function (eDoc, FAST, Data Distribution) is designed to be stand-alone module, enabling users to develop and validate one or more message sets without the hassle of validating the entire Mail.XML specification. For instance, a developer, implementing only Data Distribution messages will only validate the DD module XSDs and will ignore the eDoc and FAST XSDs.

Changes or updates in Mail.XML 11.0D specifications are **not backward compatible** with previous versions of Mail.XML for Data Distribution i.e. v11.0B, v10.0A, v8.1, 8.0B.

Mail.XML 11.0B

Mail.XML 11.0B corrects technical errors from the previous Mail.XML versions in an effort to support and update the USPS automation vision for 2012. New features were added In Mail.XML 11.0B. This document lists all of the changes for Mail.XML 11.0B version. .

The Data Distribution system will continue to support Mail.XML 10.0. In previous versions such as Mail.XML 8.1, the messages in XSDs are tightly coupled to one main XSD and one base XSD. The main XSD covers all of the message sets of the eDoc, FAST, and Data Distribution. This tight coupling requires users to validate the complete XSD. In contrast, the introduction of modularization in Mail.XML 9.0 and onwards, are designed to be stand-alone modules, which enables users to develop and validate one or more message sets without the hassle of validating the entire Mail.XML specification. For instance, a developer, implementing only the Data Distribution messages will only validate the DD XSDs and will ignore the eDoc and FAST XSDs.

Lastly, the changes or updates in Mail.XML 11.0B specifications are **not backward compatible** with previous versions of Mail.XML for Data Distribution i.e. v10.0A, v8.1.

Mail.XML 10.0

Mail.XML 10.0 is a major release that corrects technical errors from the previous Mail.XML versions and introduces the schema modularization in an effort to streamline and modernize the USPS automation vision for 2011.

The Data Distribution system will continue to support Mail.XML 8.1. In Mail.XML 8.1, the messages in XSDs are tightly coupled to one main XSD and one base XSD. The main XSD covers all of the message sets of the eDoc, FAST, and Data Distribution. This tight coupling requires users to validate the complete XSD. In contrast, the introduction of modularization in Mail.XML 9.0 and onwards, are designed to be stand-alone modules, which enables users to develop and validate one or more message sets without the hassle of validating the entire Mail.XML specification. For instance, a developer, implementing only the Data Distribution messages will only validate the DD XSDs and will ignore the eDoc and FAST XSDs.

Lastly, the changes or updates in Mail.XML 10.0 specifications are **not backward compatible** with previous versions of Mail.XML for Data Distribution i.e. v10.0A

Mail.XML 8.1

The Postal Service has made numerous changes in Mail.XML Version 8.1 to fix technical errors in previous versions. In addition, unused simple/complex types were removed, and new values were added to align the Mail.XML with the Mail.dat functionalities

General XML and Web Services Information

XML

XML stands for eXtensible Markup Language and it is designed to transport and store data. For general information on XML, please visit http://www.w3schools.com/xml/default.asp

XSD

An XSD is the XML Schema that describes the structure of an XML document. For general information on XSD, please visit http://www.w3schools.com/Schema/default.asp

Web Service

A Web Service is a feature that converts your software into internet-based software and publishes its functions/message to the users of the internet. For general information on Web Services, please visit:

http://www.w3schools.com/webservices/default.asp

http://www.w3schools.com/wsdl/default.asp

http://www.w3schools.com/soap/default.asp

About Schema Modularization

From Mail.XML 9.0 and onward publications, Mail.XML messages will be grouped into six sets:

- Transportation Messages
- Mailing Messages
- Data Distribution Messages
- Identification Messages
- Supply Chain Messages
- System Messages

The underlying schemas have been modularized to reflect these distinct message sets. Additionally, three “shared” or “common” schemas have been developed and they are:

1. One stores simple Mail.XML Mailing Message Specification type definitions that can be shared across message sets.
2. A second common module is made up of common elements and complex types that are shared across two or more message sets.
3. The final common module is made up of system messages and the attributes, elements and complex types that are *unique* to these messages.

Mail.XML schema modules are designed to stand alone, separated by business functions, enabling those interested in one or more message sets to develop and validate just those message sets without the requirement to address the entire Mail.XML specification. Each module will validate by itself or in combination with all other message sets when the entire Mail.XML Specification is considered. In addition, each module will be versioned separately.

Version numbers are contained within the namespace designation. Modules will be documented individually as well.

The following Mail.XML XSD modules/namespaces have been developed:

- **mailxml_tm.xsd:** This module contains all the transportation (or FAST) messages with their attributes, elements and complex types that are *unique* to these messages. **Namespace=mailxml_tm**
- **mailxml_mm.xsd:** This module contains all the mailing messages (eDoc) with their attributes, elements and complex types that are *unique* to these messages. **Namespace=mailxml_mm**
- **mailxml_dd.xsd:** This module contains all the data distribution messages with their attributes, elements and complex types that are *unique* to these messages. **Namespace=mailxml_dd**
- **mailxml_id.xsd**: This module contains all the identification messages (CRID, MID etc.) with their attributes, elements and complex types that are *unique* to these messages. **Namespace=mailxml_id**
- **mailxml_sc.xsd:** This module contains all the supply chain messages (Cross-functional container based and other Mail.dat/mail.XML update functions) with their attributes, elements and complex types that are *unique* to these messages. **Namespace=mailxml_sc**
- **mailxml.xsd:** This module contains all the system messages with their attributes, elements and complex types that are *unique* to these messages. Note that in order for the MessageResponseRetrievalResponse to work, all the other message modules are included by the system.xsd. **Namespace=mailxml**
- **mailxml_defs.xsd:** This module contains all the common definitions of attributes, elements and complex types that are used across one or more message types. **Namespace=mailxml_defs**
- **mailxml_base.xsd**: This module contains all the simple types that are used everywhere in the mailxml. These simple types are associated with every element of each message. **Namespace=mailxml_base**

What Schemas Cover Data Distribution Functionality?

With the breakdown of XSD into multiple schemas – the Data Distribution functionality will be covered by mailxml_dd.xsd module and mailxml_id.xsd.

In addition to the above modules, mailxml_base.xsd and mailxml_defs are common/shared XSD that will be referred by each of these individual Data Distribution specific modules.

Lastly, mailxml.xsd module will be the source XSD that will connect all messages to their system level attributes. For example – MessageResponseRetrievalResponse is shared attribute/complex type associated to almost every single Mail.XML message regardless of whether it is a Data Distribution functionality or FAST or eDoc functionality. The other examples are DeliveryResponse, NotificationResponse, Fault, etc.

USPS® Limitation Rule on Transaction Requests

USPS has recently encountered unordinary high volume with specific Mail.XML transactions by FAST and Full-Service Data Distribution users, which is causing a serious performance impact on several USPS servers that deal with authorization of all of the Mail.XML transactions.

The Postal Service is requiring that the USPS Mail.XML customers update their Mail.XML software to query the USPS *PostalOne!* FAST and Data Distribution systems **no more than seven times per hour**. Similarly, USPS customers are required to set up their web services pull software to query the *PostalOne!* servers **no more than 200 times in a given 24-hour period** for their eDoc, FAST, or Full-Service related data queries. This change is required for all customers. It will greatly assist in avoiding performance issues that can hinder quality of service; users still have many opportunities to query necessary postal data within this limitation. Likewise, in order to improve overall experience for all customers it is absolutely necessary that customers program their software to query the USPS systems for different transactions at different times, i.e., configure some queries to occur at odd times, e.g., FAST queries to occur at odd times while eDoc and Data distribution to occur at even times.

Note: *The above statements and limitations are under review by the USPS. Updates to this section will be provided in the next Mail.XML technical specification publication.*

Roadmap of Mail.XML

The table below shows the roadmap of the Mail.XML versions - starting from currently supported versions to versions supported in April 2013 for Data Distribution.

January 2012 Release 29.0	April 2012 Release 30.0	June 2012 Release 31.0	October 2012 Release 32.0	January 2013 Release 33.0	April 2013 Release 34.0
Mail.XML 10.0A Mail.XML11.0D**	Mail.XML 10.0A Mail.XML11.0D	Mail.XML 10.0A Mail.XML11.0D Mail.XML12.0A	Mail.XML11.0D Mail.XML12.0A Mail.XML12.0B	Mail.XML 11.0D Mail.XML 12.0A Mail.XML 12.0B	Mail.XML 11.0D Mail.XML 12.0A Mail.XML 12.0B

Table 0-1: Roadmap of Mail.XML Supported Versions

Note: *The version(s) flagged with double asterisk (**) are new version(s) which are planned for implementation with the release date.*

The table below lists all Mail.XML 11.0D Profiles and Full-Service Data Feedback related messages by version.

Note: *Full Service Messages such as FullServiceAddressCorrectionQuery Request or ContainerVisibilityQueryResponse have been modified in Mail.XML 11.0D by dropping the prefix "FullService" in order for these messages to support both Full and Non-Full services data structures.*

- FullServiceStartTheClockDelivery
- FullServiceStartTheClockNotification
- FullServiceStartTheClockQueryRequest
- FullServiceStartTheClockQueryResponse

These FullServiceStartTheClock messages are being removed from the scope of Mail.XML 11.0D because there exist already a similar set of StartTheClock messages except without the prefixes "FullService". These messages are being removed to avoid redundancy.

Message Name	Supported in Mail.XML 11.0D
ByForConflictDelivery	X
ByForConflictNotification	X

Message Name	Supported in Mail.XML 11.0D
ByForConflictQueryRequest	X
ByForConflictQueryResponse	X
CastofCharactersCancelRequest	
CastofCharactersCancelResponse	
CastofCharactersCreateRequest	
CastofCharactersCreateResponse	
CastofCharactersUpdateRequest	
CastofCharactersUpdateResponse	
DeliveryResponse	X
AddressCorrectionDelivery	
AddressCorrectionNotification	
AddressCorrectionQueryRequest	
AddressCorrectionQueryResponse	
ContainerVisibilityDelivery	
ContainerVisibilityNotification	
ContainerVisibilityQueryRequest	
ContainerVisibilityQueryResponse	
AddressCorrectionDelivery	X
AddressCorrectionNotification	X
AddressCorrectionQueryRequest	X
AddressCorrectionQueryResponse	X
ContainerVisibilityDelivery	X
ContainerVisibilityNotification	X
ContainerVisibilityQueryRequest	X
ContainerVisibilityQueryResponse	X
DataQualityVerificationQueryReport Request	X
DataQualityVerificationQueryReport Response	X
DataQualityVerificationReportDelivery	X
DataQualityVerificationReport Notification	X
NonComplianceDataWithOwedPostageNotification	X
NonComplianceDataWithOwedPostageDelivery	X
NonComplianceDataWithOwedPostageQueryRequest	X

Message Name	Supported in Mail.XML 11.0D
NonComplianceDataWithOwedPostageQueryResponse	X
NixieDetailDelivery	X
NixieDetailNotification	X
NixieDetailQueryRequest	X
NixieDetailQueryResponse	X
StartTheClockDelivery	X
StartTheClockNotification	X
StartTheClockQueryRequest	X
StartTheClockQueryResponse	X
NotificationResponse	X
USPSCRIDCreateRequest	X
USPSCRIDCreateResponse	X
USPSMIDCreateRequest	X
USPSMIDCreateResponse	X
MessageResponseRetrievalRequest	X
MessageResponseRetrievalResponse	X

Table 0-2: Profiles and Full-Service Data Feedback Messages in Mail.XML 11.0D

Purpose

The purpose of this Technical Specification is to supplement the IDEAlliance Mail.XML specification and to provide the Mail.XML users with specific information regarding the USPS implementation. This document serves as the collection of all specific rules, standards, accepted values, error messages, etc. regarding the USPS implementation of the Mail.XML specification. The content in this document is subject to change as IDEAlliance updates the Mail.XML Specification. This technical specification includes the return codes, error codes and their descriptions, as well the WSDLs links for the October 2012 USPS Mail.XML implementation. This specification also lists all of the return codes and error codes, along with their descriptions, for each message supported in Mail.XML 11.0D.

Intended Audience

The intended audience for this specification is technical programmers and technical managers. Prior knowledge of XML language, Mail.XML and Mail.dat functionalities will greatly improve the understanding of this document. The majority of the codes used in the Mail.XML base XSD come from the Mail.dat specification, the USPS recommends reading the *Postal Service Mail.dat Technical Specification* to understand the meaning of the flags used by IDEAlliance in Mail.XML. The code descriptions for simple types that are coming from the Mail.dat specification have been added as part of the simple type table for added benefit to the reader.

About This Document

In the messages data structure table, complex types have been referenced. To properly implement the Mail.XML messages, USPS recommends the customer study the complex types provided in Appendix A of this specification to fully understand the structure of the Full-Service Data Distribution messages. The same functionality now supports FAST push messages for stale contents to the content providers, content updated by content providers data to schedulers, and closeout data. The FAST messages are not necessarily tied to Full-Service. So from June 2011, you can subscribe to the StalecontentDelivery, StaleContentNotification, ContentUpdatedNotification, ContentUpdatedDelivery, CloseoutDelivery and CloseoutNotification messages.

Also in the messages data structure, simple types have been referenced under the acceptable values column. USPS recommends that the customer study the simple types to fully understand the cryptic meaning of the codes that are being used in the simple types. For example, “Mail Class” simple type allows values 1, 2, 3, or 4, where the translation of these codes is ‘First-Class Mail’, ‘Periodicals’, ‘Standard Mail’, and ‘Package Services’ respectively. The USPS also recommends reading the *Postal Service* Mail.dat Technical Specification document located on RIBBS®.

In the data structure tables, some field names may contain spaces for readability purposes because the names of those fields are lengthy. In Mail.XML, spaces are not allowed in the element names. For example: “SubmittingParty” is the element name but in the data table it may have been written as Submitting Party”. If you are using this document to code your software then use the latest version of Mail.XML XSD located on RIBBS® to fully understand the name of the element, structure of every block and message to minimize the impact of incorrect coding.

How to Read Data Structure

Each element listed in the data structure table of each message is designated as one of three possible types: REQUIRED, OPTIONAL, or CONDITIONALLY REQUIRED. The Required designation means that the sender (mailer/preparer) of the message must provide data for the element; it cannot be left blank. The Optional designation means the sender of the message has a choice to include the data or omit it from submission. The third option, Conditionally Required, means that if certain conditions are met, then the sender of the message must provide data for that particular element. For example, the conditionally required data element EntryPointEDPostalCode is only required if EntryPointEDFacilityType is populated in the ContainerUpdateRequest message; if EntryPointEDFacilityType is not populated, then the element EntryPointEDPostalCode is not required.

In addition to the above options, there are instances where the main block is marked as REQUIRED but the elements within the block are OPTIONAL or vice versa. This means that the sender of the message must include the block when sending the message, but can omit all elements within the block. A good example is the "SubmittingParty" block, which is a REQUIRED block, but all elements within this block are OPTIONAL.

Issue Resolution

All issues identified in the USPS software that may not correspond with the business rules identified in this specification and that require workarounds are listed in the critical issues list on the RIBBS Intelligent Mail site under Latest News. For example – In some cases the work around for Full-Service ACS data is to receive the data through OneCode ACS method instead of Full-Service method. In other cases for postage statements too big in size, the work-around is to divide the statements in multiple statements. The current Critical Issues list of the *PostalOne!* system is found at Intelligent Mail on the Latest News tab at: http://ribbs.usps.gov/intelligentmail_latestnews/documents/tech_guides/PostalOneCriticalIssueExtCustomers.pdf

Environments supported by USPS

The USPS supports a test and production environment.

Testing Environment for Mailers (TEM): This environment is used for software approval processes for all mailers prior to moving into the production environment. Mailers must work with the *PostalOne!* Help Desk to start the approval process for their software, and must be approved in the TEM environment, prior to sending Mail.XML data transactions in the Production environment. The TEM environment for Mail.XML basic testing capabilities is available as of November 2009 and will be available for all future releases. For more information regarding TEM, see the *Test Environment for Mailers: Checklist and Troubleshooting Guide*

Production Environment: This Production environment will be used by all customers approved by USPS to start sending messages and transactions. These customers will be approved as a result of performing required tests in the TEM environment.

PostalOne! WSDL Information

All web services transactions that are made with the USPS should comply with the WSDL specification, which are posted on RIBBS® and can be downloaded from the following location: http://ribbs.usps.gov/intelligentmail_guides/documents/tech_guides/xmlspec/wsdls/wsdls.htm

A web services request must include a valid and active USPS User ID and Password as defined in the UserAuthenticationInfo.xsd file that is provided as part of the WSDL files. A typical web service message construct defined is as follows:

```
<wsdl:message name="DeliveryApptContentUpdateRequest">

     <wsdl:part name="authInfo" element="auth:AuthenticationInfo"/>
    <wsdl:part name="apptContentUpdate" element="tns:apptContentUpdate"/>

</wsdl:message>
```

With the following

String-typed WSDL:

```
<wsdl:message name="StartTheClockQueryRequest">

    <wsdl:part element="auth:AuthenticationInfo" name="authInfo"/>
    <wsdl:part element="tns:startTheClockQuery" name="startTheClockQuery"/>

</wsdl:message>
```

XSD-typed WSDL:

```
<wsdl:message name="StartTheClockQueryRequest">

    <wsdl:part element="uspsmxml:StartTheClockQueryRequest"
    name="StartTheClockQueryRequest"/>

</wsdl:message>
```

USPS will validate and authenticate the user information before processing the request. An error message will be returned in response to the web service request for invalid user information.

Types of WSDLS

There are two types of WSDLs, the String-Type WSDL and Message-Type WSDL. The following is basic information on each of the WSDL types to help customers understand what are the advantages and disadvantages of consuming String versus Message Type WSDLs.

The USPS is no longer supporting the string type as of the Mail.XML 9.0 implementation. Currently, the USPS only accepts Message-Type WSDLs.

1. Consuming String-Type WSDLs:
    a. The string-type is a more generalized method, in which a customer needs to manually register each message type and register all services in their environment to later use for invoking the remote service. This is a manual process and cannot be automated easily. This is the major disadvantage of using this method. The advantage is that one string type WSDL can be used for any version of Mail.XML but is required to be manually registered in the code. The USPS no longer supports string-type WSDLs as of Mail.XML 9.0 implementation. The USPS only accepts Message-Type WSDLs.
2. Consuming Message-Type WSDLs:
    a. The message-type WSDL eliminates the need to manually register each service and message type. With this method, a customer can use an automated tool to setup their environment without manually registering each service. However, this approach requires updates to WSDLs every time a new Mail.XML version is released. In other words, it is tightly coupled with the Mail.XML Specification Version. The USPS has transitioned to this type of Message WSDLs to remove manual editing and registering of messages. USPS has stopped supporting string-type WSDLs as of the Mail.XML 9.0 implementation. It is required that all users move to the message-type WSDLs.
3. Customer must use message-type URLs to consume these services. These URLs are provided in Appendix C.

Mail.XML Workflow

Full-Service Feedback Messages Workflow

There are two ways a user receives Full-Service Feedback data, a) the Push Delivery and b) the Pull methods. All messages that end with a word 'Notification' or 'Delivery' are part of Push Delivery method and all messages that end with a word 'Request' or 'Response' are part of Pull method. To start receiving the data via Push Delivery method the user must subscribe to Push Delivery service via USPS – Business Customer Gateway accessible through the web at: https://gateway.usps.com/bcg/login.htm.

For more details on how to subscribe to Push Delivery and a more detailed description of the Pull and Push methods, refer to section 7 of this document.

Profile Management messages Workflow

All messages in the Profile Management families follow the Pull method where a user sends a request and expects to receive either data or an error/return code.

Messages, including CRID or MID creation messages, do not need to be requested or sent in any particular order.

Exceptions

The following is the list of exceptions identified by USPS.

Mail.XML Version	Exception Description
ALL	The customer should not zero pad their MID and CRID IDs

Table 0-3: Mail.XML exceptions identified by USPS

Full-Service Data Feedback and Profile Management

Full-Service Data Feedback Overview

The Full-Service program will utilize the *PostalOne!* system to provide Start-the-Clock, Container Visibility, Confirm (Full-Service only), Address Correction Data, Nixie Detail data, and Full-Service Data Quality reports to the customers identified as recipients of Full-Service feedback.

There are number of ways by which all of the Full-Service data feedback messages will be exchanged through the *PostalOne!* system as described below:

Options for Data Exchange					
No	**Data/** **Service**	**Browser-based HTTPS Download (Compressed Files)**	**Browser-based Report (over HTTPS)**	**Mail.XML Pull (Customer Invokes USPS Web Service)**	**Mail.XML Push (USPS Invokes Customer Web Service)**
1	Start-the-Clock	Yes	Yes	Yes	Yes
2	Container Visibility/Scan	Yes	Yes	Yes	Yes
3	ACS COA	Yes	No	Yes	Yes
4	ACS Nixie Detail	Yes	No	Yes	Yes
5	Full-Service eDoc Data Quality Verification	No	Yes (Microstrategy Reports)	Yes	Yes
6	Full-Service noncompliance postage	No	Yes (Microstrategy Reports)	Yes	Yes

Table 0-1: Full-Service Data Exchange Options

FTP – File Transfer Protocol method. FTP support is phasing out to meet USPS Security standards for secured, guaranteed solutions. Any USPS services that currently provide data through FTP to their customers will provide ample time to communicate and collaborate with those customers in order to help with their transition to a non-FTP data communication method.

Browser-based HTTPS (Hyper Text Transport Protocol - Secure) Download (Compressed Files) – Online web page allowing customers to download ‘compressed data’ over HTTPS protocol in a CSV or Excel format.

Browser-based HTTPS Reports – Online web page allowing customers to view ‘reports’ over HTTPS.

Mail.XML Pull (Customer Invokes USPS Web Service) -Web Services Pull model where a customer sends a Web Services-based Mail.XML Request message invoking USPS Web Services to pull data in XML format through a Mail.XML Web Service method.

On the response provided, it would include TrackingID as part of ‘LargeTransactionDividerResult’. Whenever a mailer application sees this element, they need to send MessageResponseRetrievalRequest (using the correct Mail.XML version XSD) and use the TrackingID provided to get the response.

For example, if a mailer sends ‘NixieDetailQueryRequest’ and the *PostalOne!* system has to send a large result set as a response (1000 messages broken down into 200 messages per transaction). The *PostalOne!* system will return a response with the first 200 messages and include five TrackingIDs as part of the ‘LargeTransactionDividerResult’.

Mailer can send TrackingID request and get the remaining data. Additionally, the element ‘LargeTransactionDividerGroupType’ can give more information on what is included on each transmission.

Refer to Section 2.1.5 for more information.

Mail.XML Push (USPS Invokes Customer Web Service)-Web Services Mail.XML Push model where the USPS invokes a customer’s Web Service method/URL and pushes XML data to the customer’s web service when the data is available or a notification is sent to the customer by the USPS that customer’s data is available for download. Refer to Section 2.1.4 for more information.

In support of Full-Service feedback messages that larger are in size, the Mail.XML specification has an element ‘LargeTransactionDividerResult’ to support such transactions. The *PostalOne!* system will divide the large result set into smaller chunks.

What is needed to receive data through the data exchange methods identified?

Step 1

The customers must be registered PostalOne! users and must be registered Business Customer Gateway users with a valid username and password. For information on the Business Customer Gateway, refer to the User Access to Electronic Mailing Information and Reports Guide, Volume 1 from RIBBS at

http://ribbs.usps.gov/intelligentmail_guides/documents/tech_guides/user_access/user_access.htm

Visit the Business Customer Gateway home page at https://gateway.usps.com to begin participating in Mail.XML and to obtain access to the PostalOne! system. If you have questions, call the PostalOne! systems Help Desk at (800) 522-9085.

Once you are a registered user and have the ability to access the PostalOne! system you will have access to the browser-based HTTPS reports and data download options as well, you will have the ability to perform electronic data exchange with the PostalOne! system through approved methods.

Step 2

The customers interested in automated electronic data exchange currently have four options available to them for receiving electronic data from the USPS for Full-Service mailings.

Browser-based HTTPS Download (Compressed Files) - The first available data exchange option is secure browser-based compressed file download. No other information for Full-Service will be available through browser-based download method, other than Address Correction and Nixie Detail data.

Browser-based HTTPS Reports - The second option available to customers is viewable browser-based reports. The only data that is currently available through browser-based reports is Start-The-Clock and Container Visibility data.

Mail.XML Pull (Customer Invokes USPS Web Services) – The Mail.XML is an IDEAlliance specification and does not require any IDEAlliance fee to use/implement the specification. The latest version of Mail.XML 11.0D is available at the http://www.mailxml.org site. Customers can pull their information whenever they desire, using Request and Response methods for StartTheClock, Container Scans, ACS, and Nixie data.

Mail.XML Push (USPS Invokes the Customer's Web Services) – USPS provides online push subscription pages as well as web services methods so that the customer can register with the *PostalOne!* system as a subscriber of detail data (Delivery messages) or as a subscriber of notifications (Notification messages) that tell the customer, when data is ready to be pulled. The customer can register to receive automatically, either a) the detail data, or b) the notifications as they pertain to StartTheClock, Container Scans, Full-Service ACS COA, and Full-Service ACS Nixie data.

The Mail.XML specification is a USPS/industry agreed to strategic and future communication method for data exchange. It is bi-directional, near real time and can convert large data transactions into smaller transactions while providing guaranteed delivery, verification and near real time conflict resolution capabilities. The sender and receiver can validate the data, verify guaranteed delivery and actually converse intelligently both ways to report any issues with data transfer or with business issues found within the data.

How to implement available Mail.XML versions for Full-Service Data Feedback?

The user should download the XSD, the data dictionary and support documents from the following link. http://www.mailxml.org.

Check for the versions supported by the USPS at http://ribbs.usps.gov/intelligentmail_guides/documents/tech_guides/xmlspec/wsdls/wsdls.htm under Intelligent Mail Services -> Guides links.

Why Mail.XML based Data Feedback?

Mail.XML version 6.0D was created to enhance efficiency and ease of transmission for the small and large data volume communications. The message sets were designed with an attribute that allows all large data volume messages to be broken up in as many unique child transactions as the USPS and the industry feel will be supported by the existing network infrastructure. In other words, a 10 MB file can be broken up in ten 1MB transactions or twenty 500KB in size transactions except for Address Correction and Nixie messages where the messages will be broken between 450-600KB. The Mail.XML allows for verification and thorough validation of the data, right away, in near real time. More importantly, with Mail.XML, your software can respond back to the *PostalOne!* system right away to let the USPS know if you successfully received all the data or not. You can also let the USPS know, right away, if the data sent by the USPS to you had problems, the data was bad or incomplete or did not belong to you, etc., so automatic conflict resolution triggers can result in immediate action by the USPS. Mail.XML 7.0C expands the same capabilities to other data types. Mail.XML 8.0B adds more validation by introducing changes to simple types and Mail.XML 8.1 further streamlines the XSD structure to fix the previously identified exceptions such as making the SubmittingParty and other elements as optional in all data distribution messages. Starting from Mail.XML 9.0 versions and onward – modularization of XSD has been introduced which will streamline the process of implementing the code.

Other important thing for your business is that unlike Mail.dat, there are almost no wait times for you to file your postage, qualification, and container and bundle reports with the USPS since you are filing final documents.

Prerequisite: All customers who want to receive Full-Service data feedback must be registered *Business Customer Gateway* users. The customer must have gone through the *Gateway* approval and activation process for this business function and must establish push subscription profiles in the Push Subscription user interface to provide data distribution guidance for their data. The data distribution authorization is driven by the data provided in the eDocumentation to the *PostalOne!*

system about the By/For relationship. The MID profiles if setup by the customer, take precedence on authorization rules over the eDoc data.

The Full-Service data is provided electronically to the USPS customers in two ways. One method is through manual download of data in CSV (Comma Separated Values) and Microsoft Excel application formats available through Business Customer Gateway online pages. Second method is of an automated data exchange, using IDEAlliance Mail.XML specification and web services technology. With Mail.XML method, the customer can periodically query the USPS *PostalOne!* system for their information or can subscribe for their data and receive it automatically from the USPS *PostalOne!* server to their (customer's) server. Mail.XML based data exchange requires that the customer buy software that is capable of exchanging information with the USPS through SOAP protocol or that the customer code (programs) the software with Mail.XML capabilities.

Customers can login through Business Customer Gateway, go to Full-Service reports link to search for specific ACS and StartTheClock data for their mailings, and download the data in a CSV or Excel format.

> ***Note:*** *The Data Quality reports are available through manual download format, online reports, and through Mail.XML format, however we do not encourage manual download of this detail data because the data would only make sense once merged/linked with the mailing eDoc data in customer's systems.*

Customers also have the ability to conduct all Full-Service data exchange electronically and automatically through IDEAlliance Mail.XML web services implementation by the USPS.

What is Push Model/Subscription method?

If you want to automatically receive the data as soon as it is ready (Push model), you must have registered your Web Services URL, implemented a WSDL for push (Delivery and Notification) Mail.XML messages, and implemented Authentication and Authorization to authenticate and authorize transactions sent by the *PostalOne!* system to your server. You also must have installed a signed SSL certificate from VeriSign or Comodo on your server, which is setup to receive push transactions form the USPS. You will also be asked to provide your IP address, server's physical address and other pertinent information to the Help Desk in the TEM Full-Service Data Feedback using Push Messages – Customer Identification Form that can be downloaded from RIBBS at

http://ribbs.usps.gov/intelligentmail_guides/documents/tech_guides/xmlspec/xmlspec.htm

With the Mail.XML Push or subscription method, the USPS customers can login to the USPS Business Customer Gateway and under Full-Service subscription link can provide a user id, password, and a URL that the *PostalOne!* system can use to connect and send data to the customer's server.

On the Business Customer Gateway, the USPS customer can schedule how often they want to receive their Full-Service data. They have the choice to get the data every 6, 12, or 24 hours.

The WSDLs and XSDs available at the end of this document MUST be used to ensure conformity and standardization of data exchange between the *PostalOne!* system and customers. Within the Gateway system, a user interface allows you to provide the information and indicate which messages you want to receive as a Push model. You must have also checked the type of data you want to receive as a Push. All push messages **end** in the word "Delivery" or in the word "Notification." Messages that **start** with the word "Delivery" are usually specific to the FAST system.

The messages that end with the word "Delivery" mean that the message has all the detail data and the customer is receiving the data because the customer registered with the USPS *PostalOne!* system to receive the detail data as it becomes available.

The messages that end with the word “Notification” mean that the message will notify the customer that the “data is ready” to be pulled through Pull Mail.XML mechanism or another approved method. Notification messages will not have detail data. The customer will have the option to choose the push messages that they want the *PostalOne!* system to use and send the data to the customer. e.g., a customer may decide to use Pull messages for ACS, but may decide to receive continuous feeds of data for StartTheClock “Delivery” message or Nixie “Delivery” message and may choose to receive the notifications through the Notification messages.

Getting Ready for Push Data Distribution Checklist

The details about ‘How to Subscribe to Mail.XML Push Messages’ are outlined in the ‘Test Environment for Mailers (TEM) Checklist and Troubleshooting document on RIBBS. In this section – high-level steps are provided for an overall understanding of the workflow:

1. Enable SSL communication with USPS® TEM and Production environment. *Refer to the ‘Test Environment for Mailers (TEM) Checklist and Troubleshooting* document*.*
2. Customer must acquire signed certificate to enable SSL (secure socket layer) communication with USPS®, which allows secure data exchange between USPS® and customers server on port 443.
3. Through Business Customer Gateway – access the Push Subscription Profile user interface and setup Subscription profile
   a. During Subscription profile setup – user has two options i.e. Subscribe to Notification and Subscribe to Delivery. Customer can subscribe to one push type.
   b. During the subscription setup process, the customer must determine how frequently they would like to receive the notification or delivery.
   c. Customer must provide their servers’ web service URL, User Name, and Password. This will allow the USPS *PostalOne!* system to connect to customers’ servers and start pushing the notification or deliver the message.
4. Customer must setup their Web Service to access USPS® TEM and USPS® Production environment and provide the Web Service URL during the subscription process.
5. Download and implement USPS® provided WSDLs. *See WSDL Information below for more details*

What is Pull Model?

If a customer wants to get the data based upon their own needs and wants to pull the data from the *PostalOne!* system, they will not need to host their own Web Services URL. They will need the software that can send a message request to the *PostalOne!* system for any available data and the *PostalOne!* system will respond back with the data. The customer will need to be approved and activated by the USPS to be able to use this software in production.

With the Mail.XML Pull method, the USPS customers can have their software query the USPS *PostalOne!* System for their ACS, Nixie, Start-the-Clock, Container-Visibility, By-For-Conflict, Non-Compliance Data, and Data Quality reporting data.

Message Response Retrieval Request (MRRR) Processing

Fault and Fault Recovery Messages

This series of messages is primarily used to support the Asynchronous communication when the response from the USPS may not be within a 7 seconds period and the USPS gives out a tracking ID to the customer. The customer then uses the Tracking ID to get his message response back using the MRRR message set that includes:

Message Response Retrieval Request

Message Response Retrieval Response

Customer will have the ability to retrieve the responses of all of the messages documented in this specifications using TrackingID in the MRRR messages. Tracking IDs become stale after 24 hours.

Error/ Return Codes

Faults, such as a message timeout or invalid XML are to be communicated using the fault element and returned within the detail section of the SOAP fault. The fault is made up of a tracking ID and one or more fault codes and optional fault descriptions. Note all of the Error/Return Codes listed below are supported in all versions of the Mail.XML.

Below is the list of fault codes that are applicable to all Mail.XML messages, they are sent whenever a fault occurs in the transmission of the message.

Code	Description	Messages
402	Not Well Formed XML	Fault (All Mail.XML Messages)
403	Validation Failure – {specific error message thrown by parser}	Fault (All Mail.XML Messages)
412	Unauthorized – Required SubmittingParty	Fault (All Mail.XML Messages)
412	Unauthorized – User does not have access to specified SubmittingParty	Fault (All Mail.XML Messages)
412	Unauthorized – User does not have access to “Manage Mailing Activity” service	Fault (All Mail.XML Messages)
412	Unauthorized - Invalid user ID	Fault (All Mail.XML Messages)
412	Unauthorized - Invalid user password	Fault (All Mail.XML Messages)
412	Unauthorized - Account disabled	Fault (All Mail.XML Messages)
412	Unauthorized - Maximum password retries reached	Fault (All Mail.XML Messages)
412	Unauthorized - Profile not found	Fault (All Mail.XML Messages)
440	Sorry - Mail.XML version is not supported	Fault (All Mail.XML Messages)
500	Generic Internal Responder Error	Fault (All Mail.XML Messages)
500	Your request has been accepted for processing by USPS. Use the attached Tracking ID with a MessageResponseRetrievalRequest message to get the status of your request.	Fault (All Mail.XML Messages)
500	Generic Internal Responder Error (Authentication Service Failed)	Fault (All Mail.XML Messages)
503	Not Implemented - Mail.XML message received is not supported (MessageName)	Fault (All Mail.XML Messages)

Table 0-2: Fault Codes

Below is the list of the codes to be returned by the Customer to the *PostalOne!* system in the Delivery Response message to indicate the success or the failure of the delivery messages to the Customer's web services end point

Code	Description	Messages
5800	Successful Push Delivery Receipt	DeliveryResponse NotificationResponse
5801	Push Authentication failed: Invalid Username	DeliveryResponse NotificationResponse
5802	Push Authentication failed: Incorrect Password	DeliveryResponse NotificationResponse
5803	Push Account Disabled	DeliveryResponse NotificationResponse
5804	Push Delivery Processing Failed	DeliveryResponse NotificationResponse

Table 0-3: *PostalOne!* - Data Distribution Delivery Response Return Code Values

Below is the list of the return codes that the *PostalOne!* system will communicate to the customer in response to the Push Subscription request message to indicate the success or the failure of the subscription request.

Code	Description	Messages
5805	Push Subscription Error: Invalid CRID.	ProfileResponse
5806	Push Subscription Error: Invalid message type.	ProfileResponse
5807	Push Subscription Error: Invalid Push type.	ProfileResponse
5808	Push Subscription Error: Invalid user ID and/or password.	ProfileResponse
5809	The web service URL format is invalid.	ProfileResponse
5810	Push Subscription Error: Push Start date is invalid.	ProfileResponse
5811	Push Subscription Error: Invalid frequency.	ProfileResponse
5812	Push Subscription created successfully.	ProfileResponse
5813	Push Subscription updated	ProfileResponse

Code	Description	Messages
	successfully.	
5814	Push Subscription cancelled successfully.	ProfileResponse
5815	Push Subscription Error: Database error occurred, subscription not saved/updated.	ProfileResponse
5816	Push Subscription Error: Database error occurred, subscription already exists.	ProfileResponse
5817	Push Subscription Error: Database error occurred, subscription does not exist.	ProfileResponse
5900	No records found for the given search criteria	AddressCorrectionQueryResponse ContainerVisibilityQueryResponse DataQualityVerificationReportQueryResponse NixieDetailQueryResponse StartTheClockQueryResponse ByForConflictQueryResponse
5901	Full-Service data feedback has been deactivated to one or more Mail Owner CRIDS	AddressCorrectionQueryResonse AddressNixieDetailQueryResponse
5902	Please limit the date range in the search criteria to one week	AddressCorrectionQueryResonse AddressNixieDetailQueryResponse StartTheClockQueryResponse
5903	Either Mailer ID6 or Mailer ID9 is required to use lower/upper IMB ranges in search criteria	AddressCorrectionQueryResonse AddressNixieDetailQueryResponse
1500	Invalid Tracking ID	AddressCorrectionQueryResponse ContainerVisibilityQueryResponse DataQualityVerificationReportQueryResponse NixieDetailQueryResponse StartTheClockQueryResponse ByForConflictQueryResponse
1502	Come Back Later	AddressCorrectionQueryResponse ContainerVisibilityQueryResponse DataQualityVerificationReportQueryResponse

Code	Description	Messages
		NixieDetailQueryResponse StartTheClockQueryResponse ByForConflictQueryResponse

Table 0-4: *PostalOne!* Subscription Request Return Code Values

Below is the list of the return codes that the *PostalOne!* system will communicate to the customer in response to the Cast of Characters Create, Cast of Characters Update, or Cast of Characters Cancel message to indicate the failure of a request message:

Code	Description	Message
6000	Success	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6001	Invalid Mailer ID	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6002	Missing Required Field	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6003	Invalid Participant ID	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6004	Invalid Submitting CRID ID	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6005	For Future Use	
6006	Submitting Software Version is not supported	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6007	Unknown Error	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6008	Mail XML version is not supported	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse

Code	Description	Message
6009	Mailing Piece Group Range does not exist	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6010	For Future Use	
6011	Cannot update record, original record does not exist	CastofCharactersCancelResponse
6012	Cannot cancel record, original record does not exist	CastofCharactersCancelResponse
6013	XML Not well formed	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6014	Data distribution profile has not been established by the mailer	CastofCharactersCreateResponse
6015	Invalid Mailing Piece Group Range which overlaps with an existing profile	CastofCharactersUpdateResponse
6016	Cannot update record, mailpiece group has not been defined previously	CastofCharactersUpdateResponse
6017	Cannot perform request, mailer is blocked	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6018	Invalid Mailer ID in IMb	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6019	Invalid Mailer ID in Piece Range	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6020	Invalid Submitting Mailer ID	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6021	Invalid Participant Mailer ID	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse
6022	Inactive Mailer ID in IMb	CastofCharactersCreateResponse, CastofCharactersCancelResponse CastofCharactersUpdateResponse

Table 0-5: *PostalOne!* – Cast of Characters Response Return Code Values

Below is the list of return codes that the *PostalOne!* System will communicate to the customer in response to Data Distribution Messages:

ERROR CODE	REPORT TYPE	FUNCTIONAL DESCRIPTION
7106	By/For Verification	The Mail Preparer was not identified for the physical piece due to an invalid Mailer ID(.mpa)
7107	By/For Verification	The Mail Preparer was not identified for the physical piece due to an invalid CRID(.mpa)
7108	By/For Verification	The Mail Preparer was not identified for the physical piece due to an invalid Mailer ID(QualReport)
7109	By/For Verification	The Mail Preparer was not identified for the physical piece due to an invalid CRID(QualReport)
7110	By/For Verification	The Mail Preparer was not identified for the physical piece due to no Mail Preparer identifiers in the eDoc
7111	By/For Verification	The Mail Owner was not identified for the physical piece due to an invalid Mailer ID(.cpt)
7112	By/For Verification	The Mail Owner was not identified for the physical piece due to an invalid Mailer ID(.mpa)
7113	By/For Verification	The Mail Owner was not identified for the physical piece due to an invalid CRID(.cpt)
7114	By/For Verification	The Mail Owner was not identified for the physical piece due to an invalid CRID(.mpa)
7115	By/For Verification	The Mail Owner was not identified for the physical piece due to an invalid Mailer ID(MailPiece)
7116	By/For Verification	The Mail Owner was not identified for the physical piece due to an invalid CRID(MailPiece)
7117	By/For Verification	The Mail Owner was not identified for the physical piece due to an invalid CRID(QualReport)
7118	By/For Verification	The Mail Owner was not identified for the physical piece due to no Mail Owner identifiers in the eDoc
7119	By/For Verification	The Mail Owner was not identified for the physical piece due to invalid permit information(.mpa)
7120	By/For Verification	The Mail Owner was not identified for the physical piece due to invalid permit information(MailPiece)
7121	By/For Verification	The Mail Preparer was not identified for the physical piece due to invalid permit information(QualReport)

7122	By/For Verification	The Mail Preparer was not identified for the piece range due to invalid permit information(QualReport)
7123	By/For Verification	The Mail Owner was not identified for the physical piece due to an invalid publication number(.mpa)
7124	By/For Verification	The Mail Owner was not identified for the physical piece due to an invalid publication number(MailPiece)
7125	By/For Verification	The Mail Preparer was not identified for the physical piece due to an invalid publication number(QualReport)
7126	By/For Verification	The Mail Preparer was not identified for the piece range due to an invalid publication number(QualReport)
7127	By/For Verification	The Mail Preparer was not identified for the piece range due to an invalid Mailer ID(.imr)
7128	By/For Verification	The Mail Preparer was not identified for the piece range due to an invalid CRID(.imr)
7129	By/For Verification	The Mail Preparer was not identified for the piece range due to an invalid Mailer ID(PieceRange)
7130	By/For Verification	The Mail Preparer was not identified for the piece range due to an invalid Mailer ID(QualReport)
7131	By/For Verification	The Mail Preparer was not identified for the piece range due to an invalid CRID(PieceRange)
7132	By/For Verification	The Mail Preparer was not identified for the piece range due to an invalid CRID(QualReport)
7133	By/For Verification	The Mail Preparer was not identified for the piece range due to no Mail Preparer identifiers in the eDoc
7134	By/For Verification	The Mail Owner was not identified for the piece range due to an invalid Mailer ID(.imr)
7135	By/For Verification	The Mail Owner was not identified for the piece range due to an invalid CRID(.imr)
7136	By/For Verification	The Mail Owner was not identified for the piece range due to an invalid Mailer ID(Piece Range)
7137	By/For Verification	The Mail Owner was not identified for the piece range due to an invalid CRID(PieceRange)
7138	By/For Verification	The Mail Owner was not identified for the piece range due to an invalid CRID(QualReport)
7139	By/For Verification	The Mail Owner was not identified for the piece range due to no Mail Owner identifiers in the eDoc
7239	CSA Verification	The CSA ID was not provided in the eDoc(.seg/.csm or QualReport) for the Origin entered Container when a First Class CSA exists in FAST based on the Mail Facility ID and Processing Category

7240	CSA Verification	The CSA from the eDoc(.seg/.csm or QualReport) is invalid or a match cannot be found in FAST for the associated container record(.csm or QualReport)
7242	CSA Verification	The CSA associated to an eDoc Container record(.csm or QualReport) does not have an Active status in FAST
7243	CSA Verification	The eDoc Scheduled Ship/Induction Date(.csm or QualReport) is before the CSA Effective Date in FAST for the associated container record(.csm or QualReport)
7244	CSA Verification	The eDoc Principal Processing Category(.seg or QualReport) does not match the CSA Processing Category in FAST for the associated eDoc container record(.csm or QualReport)
7245	CSA Verification	The eDoc Container Level Code(.csm or QualReport) does not match the CSA Processing Code in FAST for the associated eDoc container record(.csm or QualReport)
7246	CSA Verification	The eDoc Container Destination ZIP(.csm or QualReport) does not match the CSA Container Destination ZIP in FAST for the associated eDoc container record(.csm or QualReport)
7247	CSA Verification	The eDoc Entry Point Locale Key(.csm or QualReport) does not match the CSA Facility Locale Key in FAST for the associated eDoc container record(.csm or QualReport)
7248	CSA Verification	The eDoc Scheduled Ship/Induction Date(.csm or QualReport) does not match the CSA Day of Week in FAST for the associated eDoc container record(.csm or QualReport)
7301	Mailer ID Verification	The Mailer ID in the IMcb from the eDoc container record (.csm or QualReport) is invalid or cannot be found for the associated container.
7302	Mailer ID Verification	The Mailer ID in the IMtb from the eDoc handling unit record (.csm or QualReport) is invalid or cannot be found for the associated container
7304	Mailer ID Verification	The Mailer ID in the IMb from the eDoc piece record (.pdr or MailPiece) is invalid or cannot be found for the associated piece
7303	Mailer ID Verification	The Mailer ID in the Piece Range Serialization from the eDoc piece range (.imr or PieceRange) is invalid or cannot be found for the associated piece range
7401	Service Type ID Verification	The Service Type Identifier in the IMb from the eDoc piece record (.pdr or MailPiece) is invalid
7402	Service Type ID Verification	The Service Type Identifier in the IMb from the eDoc piece record (.pdr or MailPiece) indicates the piece is Basic for a Full-Service Piece as part of a Full-Service or Mixed Job
7403	Service Type ID Verification	The Service Type Identifier in the IMb from the eDoc piece record (.pdr or MailPiece) indicates a Mail Class that does not match the Mail Class in the eDoc (.seq or QualReport)

7404	Service Type ID Verification	The Service Type Identifier in the IMb from the eDoc piece range record (PieceRange) is invalid
7405	Service Type ID Verification	The Service Type Identifier in the IMb from the eDoc piece range record (PieceRange) indicates the piece is Basic for a Full-Service Piece as part of a Full-Service or Mixed Job
7406	Service Type ID Verification	The Service Type Identifier in the IMb from the eDoc piece range record (PieceRange) indicates a Mail Class that does not match the Mail Class in the eDoc (QualReport)
7502	Unique Container Barcode Verification	The same Container Barcode(IMcb) from the eDoc(.csm or QualReport) was used more than once within 45 days from the Postage Statement Mailing Date, across Jobs and Mail Facility IDs
7503	Unique Container Barcode Verification	The same Container Barcode (IMcb) from the eDoc (.csm or QualReport) was used more than once within 45 days from the Postage Statement Mailing Date, within the same Job and Mail Facility ID
7504	Unique Container Barcode Verification	The same Container Barcode (IMcb) from the eDoc (.csm or QualReport) was used more than once within 45 days from the Postage Statement Mailing Date, across Jobs with the same Mail Facility ID
7605	Unique Handling Unit Barcode Verification	The same Tray Barcode(IMtb) from the eDoc(.csm or QualReport) was used more than once within 45 days from the Postage Statement Mailing Date, across Jobs and Mail Facility IDs
7603	Unique Handling Unit Barcode Verification	The same Tray Barcode (IMtb) from the eDoc (.csm or QualReport) was used more than once within 45 days from the Postage Statement Mailing Date, within the same Job and Mail Facility ID
7604	Unique Handling Unit Barcode Verification	The same Tray Barcode (IMtb) from the eDoc (.csm or QualReport) was used more than once within 45 days from the Postage Statement Mailing Date, across Jobs with the same Mail Facility ID
7703	Unique Piece Barcode Verification	The same Piece Barcode(IMb) from the eDoc(.csm or MailPiece) was used more than once within the 45 days from the Postage Statement Mailing Date, across Jobs and Mail Facility IDs
7704	Unique Piece Barcode Verification	The same Piece Barcode (IMb) from the eDoc (.pdr or MailPiece) was used more than once within 45 days from the Postage Statement Mailing Date, within the same Job and Mail Facility ID
7705	Unique Piece Barcode Verification	The same Piece Barcode (IMb) from the eDoc (.pdr or MailPiece) was used more than once within 45 days from the Postage Statement Mailing Date, across Jobs with the same Mail Facility ID

7706	Unique Piece Range Barcode Verification	The same Piece Range Serialization(IMb) from the eDoc(.imr or PieceRange) was used more than once within the 45 days from the Postage Statement Mailing Date, across Mail Facility IDs
7707	Unique Piece Range Barcode Verification	The same Piece Range Serialization(IMb) from the eDoc(.imr or PieceRange) was used more than once within 45 days from the Postage Statement Mailing Date, within the same Job and Mail Facility ID
7708	Unique Piece Range Barcode Verification	The same Piece Range Serialization(IMb) from the eDoc(.imr or PieceRange) was used more than once within 45 days from the Postage Statement Mailing Date, across Jobs with the same Mail Facility ID
7901	Delivery Point Verification	IMb does not have a valid 11, 9, or 5 digit delivery point or the 11, 9, or 5 digit delivery point was not active within allowable period of time from the postage statement mailg date
7902	Delivery Point Verification	IMb has a value of '0000' within positions 6-9 of the delivery point
7903	Delivery Point Verification	IMb has a value of '9999' within positions 6-9 of the delivery point and the address record type is not 'General Delivery'
7904	Delivery Point Verification	IMb within the range does not have a valid 11, 9, or 5 digit delivery point or the 11, 9, or 5 digit delivery point was not active within allowable period of time from the postage statement mailg date
7905	Delivery Point Verification	IMb within the range has a value of '0000' within positions 6-9 of the delivery point
7906	Delivery Point Verification	IMb within the range has a value of '9999' within positions 6-9 of the delivery point and the address record type is not 'General Delivery'
7602	Default IMtb Code Error	The eDoc tray barcode(.csm or QualReport) provided is 24 characters and consists entirely of the digit 9 (i.e. 999999999999999999999999)
7016	Entry Facility Verification	The entry facility identified by the Locale Key within the container was unknown in FDB
7017	Entry Facility Verification	The entry facility identified by the Postal Code within the container was unknown in FDB
7018	Entry Facility Verification	The entry facility identified by the Locale Key within the orphan handling unit was unknown in FDB
7019	Entry Facility Verification	The entry facility identified by the Postal Code within the orphan handling unit was unknown in FDB
7020	Service Performance Verification	The Postage Statement Finalization Date is 2 or more days after the Scheduled Ship Date for the DMU Verified, USPS Transported Physical Container

7021	Service Performance Verification	The Postage Statement Finalization Date is 2 or more days after the Scheduled Ship Date for the DMU Verified, USPS Transported Physical Handling Unit
7022	Service Performance Verification	The Orphan Handling Unit does not have an induction method of BMEU.

Table 0-6: *PostalOne!* – Mail Quality Verification Response Return Code Value

Fulfillment of Unassociated ACS records

Currently, when ACS generates COA and Nixie records with a Full Service STID, these records are sent to SASP for fulfillment. SASP attempts to associate these records to electronic documentation for a period of 35 days. If the ACS records are unable to be associated within the 35-day period, SASP discards the records.

SASP sends reports for records with electronic documentation to the *PostalOne!* system. Additionally, for ACS records that are unable to be associated within the 35-day period, SASP determines the mail owner and data recipient CRID based on the MID in the IMb, determine the uniqueness of each unassociated mailpiece, and send the records to the *PostalOne!* system for fulfillment.

SASP calculates charges for unassociated and multiple notification records and sends ACS billing information. This information is used to create Mailer paper invoices. Within the *PostalOne!* system, mail owners and their delegates will be able to access the ACS records generated. The *PostalOne!* system supports Full Service ACS push/pull requests/responses, notifications, and downloads for mailers with no associated electronic documentation.

ACS Billing

Currently ACS does not bill for records that include a Full Service STID. These records are fulfilled for free through the Full Service platform.

ACS updates the mailpiece records with billing data after receiving the billing files from SASP. As per established ACS billing policies, ACS will generate monthly invoices to send to mailers when account balances are over a configurable value (currently set to $50); otherwise, invoices will be sent to mailers on March 24th or October 24th.

ACS will send a monthly file to Customer Registration containing all CRIDs to be invoiced. Customer Registration will use this information to retrieve billing contact information for the BSA of managed mailing activity based on the mail owner CRID and will send this information back to ACS in order to generate invoices.

Disabling Full Service Data Feedback for Delinquent Accounts

Currently, the only established method for disabling mailers’ access to their ACS records is to manually deactivate their MIDs in the ACS system. The account of any mail owner that does not make a payment within 30 days of the invoice creation date will be considered a delinquent account. Implementation of a new interface between SASP and the PostalOne! system will update a mail owner's delinquent account status.

ACS sends mail owner CRIDs for delinquent accounts to the PostalOne! system, which will deactivate the mail owner’s access to Full Service ACS feedback. Accounts can be reactivated when ACS sends activation records for mail owner CRIDs after a payment has been received.

General Considerations when using Full-Service Query Request messages

These messages allow USPS to support unassociated ACS records and to provide flexibility for customers, especially third party agents, to access and retrieve Full-Service data feedback. Starting with the IDEAlliance Mail.XML 8.1 specification, Job ID and Mailing Group ID were made optional in the ACS and NIXIE Full-Service Pull requests. These changes started being supported in November 2010.

With all key fields in Full-Service ACS and NIXIE requests annullable in the Mail.XML 8.1 Specification, there is a chance that very huge datasets may be returned which could potentially cause some performance and timeout issues. Mailers are expected to provide one of the following fields in the Mail.XML request message (Mail.dat Job ID/User License Code or Mailing Group ID or MID on Piece) and are required to limit the date range search to one week only.

Message Choreography for Data Distribution

No	Message Name	Description	Customer	USPS PostalOne!
1	StartTheClockQueryRequest	The customer requests the Start-the-Clock data by providing appointment and job information for which Start-the-Clock data is needed. The date range of upper and lower is necessary to receive data from the USPS.	From	
2	StartTheClockQueryResponse	The USPS responds with the Start-the-Clock data for the job and provides the data for plant loaded mail in the plan loaded XML block, data for drop Shipped mail in the drop shipped XML block, and BMEU entered mail data in the BMEU XML block.		From
3	ContainerVisibilityQueryRequest	The customer can request container visibility information from the USPS by providing the job information, appointment/trip information, facility id, and container information or container barcodes.	From	
4	ContainerVisibilityQueryResponse	The USPS responds with the Container Induction detail data by providing scan dates and times as well as any conflicts with the container back to the customer.		From
5	AddressCorrectionQueryRequest	The customer can request address correction information from the USPS by providing the job information and date ranges and optional piece barcode range information.	From	
6	AddressCorrectionQueryResponse	The USPS, in response to the requests, sends all the address correction data pertinent to the request.		From
7	NixieDetailQueryRequest	The customer can request Nixie information from the USPS by providing the CRID, MID, and by providing the job information, class of mail, date ranges and optional piece barcode range information.	From	
8	NixieDetailResponse	The USPS, in response to the requests, sends all the Nixie detail data by MID and Job information back to the customer.		From

No	Message Name	Description	Customer	USPS PostalOne!
10	StartTheClockNotification	The USPS sends this notification to the customer and provides the Job information for which the Start-the-Clock data is ready		From
11	StartTheClockDelivery	The USPS sends this detail data delivery to the customer with all the data available in the StartTheClockQueryResponse message		From
12	ContainerVisibilityNotification	The USPS sends this notification to the customer and provides the Job information for which the ContainerVisibility data is ready		From
13	ContainerVisibilityDelivery	The USPS sends this detail data delivery to the customer with all the data available in the ContainerVisibilityQueryResponse message		From
14	AddressCorrectionNotification	The USPS sends this notification to the customer and provides the Job information for which the AddressCorrection data is ready		From
15	AddressCorrectionDelivery	The USPS sends this detail data delivery to the customer with all the data available in the AddressCorrectionQueryResponse message		From
16	NixieDetailNotification	The USPS sends this notification to the customer and provides the Job information for which the NixieDetail data is ready		From
17	NixieDetailDelivery	The USPS sends this detail data delivery to the customer with all the data available in the NixieDetailQueryResponse message		From
18	NonComplianceDataWithPostageOwedQuery Notification	The USPS sends this notification to the customer providing information on data availability		From
19	NonComplianceDataWithPostageOwedQuery Delivery	The USPS send the detail data to customer server		From
20	NonComplianceDataWithPostageOwedQuery Request	The USPS customer constructs the message and send it to USPS servers	From	

No	Message Name	Description	Customer	USPS PostalOne!
21	NonComplianceDataWithPostageOwedQuery Response	The USPS responds to the Query Request message with either data or error code through this message.		From
22	NotificationResponse	This message is used in response to all messages that end in the word ‘Notification’	From	
23	DeliveryResponse	This message is used in response to all messages that end in the word ‘Delivery’	From	

Table 0-1: Message Choreography for Data Distribution

Mail.XML 11.0 Data Feedback and Profile Management and Miscellaneous Messages

The section below lists all messages that will be supported in the Mail.XML 11.0 specification. This section spells out the business rules and data structure for each of those messages.

ByForConflictNotification

This message is sent by USPS to customer end server notifying them with the availability of By/For conflict data

Prerequisites

- Mailer has subscribed to receive By/For Conflict Notification messages
- Mailer submits Mail.dat jobs or Mail XML eDoc messages to *PostalOne!*
- USPS checks the availability of By/For Conflict data about discrepancy within the eDoc at different levels for the owner and preparer information.
- USPS pushes the notification message to the Mailer's web service end point

Business Rules

If the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide a date of notification and will provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard to the customer.

ByForConflictNotification Data Structure

Field Description

ByForConflictNotification				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
SubmittingParty	ParticipantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	SubmittingSoftware complex type	-	Optional	Refer to this complex type in Appendix A
PushMessageID	String 25	-	Optional	-

ByForConflictNotification				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Choice Block BEGINS			Optional choice block. Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID can be provided.	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Required when Mail.XML version is used	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
FullServiceAvailabilityDate	Date	yyyy-mm-dd	Required	-
NotificationDate	Date	yyyy-mm-dd	Required	-

ByForConflictDelivery

This message is pushed by USPS to customer end server providing By/For conflict data

Prerequisites

- Mailer has subscribed to receive ByForConflictDelivery messages
- Mailer has subscribed to receive By/For Conflict Notification messages
- Mailer submits Mail.dat jobs or Mail XML eDoc messages to the *PostalOne! system.*
- USPS checks the availability of By/For Conflict data about discrepancy within the eDoc at different levels for the owner and preparer information.
- USPS pushes the notification message to the Mailer’s web service end point

- USPS pushes ByForConflictDelivery message to the Mailer's web service end point

Business Rules

If the customer has registered with the *PostalOne!* system to receive the By/For Conflict Delivery messages, the *PostalOne!* system will provide the By/For Conflict detail data to the customer about By/For Conflicts.

ByForConflictDelivery Data Structure

Field Description

ByForConflictDelivery				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
LargeTransactionDividerGroup OptionalType	LargeTransactionDividerGroup OptionalType Attribute block	-	Required	Refer this attribute block in Appendix A
SubmittingParty	participantID Type complex type		Optional	Refer to this complex type in Appendix A
SubmittingSoftware	SubmittingSoftware complex type		Optional	Refer to this complex type in Appendix A
DataRecipient	dataRecipient Complex Type	-	Optional	Refer to this complex type in Appendix A
PushMessageID	String 25	-	Optional	
ByForConflict	byForConflict Type complex type	-	Required 1 to many allowed	Refer to this complex type in Appendix A

ByForConflictQueryRequest

This message is sent by customer to pull By/For conflict data

Prerequisites

- Customer uses Mail.XML 11.0D spec
- USPS validâtes Mail.XML message
- USPS retrieves the data based on the search criteria provided in the request
- USPS returns response to the Customer

Business Rules

With each ByForConflictQueryRequest message, the customer needs to provide information for all required blocks.

1. SubmittingParty – For all data distribution for full-service messages (i.e., non-FAST, non-eDoc messages), the *PostalOne!* system will only accept a MID or a CRID as a valid ID.
2. SubmittingSoftware – For Mail.XML messages that it receives from the customer, the *PostalOne!* system will always require:
   a. A valid software name in the SoftwareName element
   b. A valid software vendor company name in the vendor element
   c. The most current version number of the software in the Version element
   d. A valid Approval Date and a valid Approval Key in the ApprovalDate and ApprovalKey elements. The approval date is the date when the USPS approved the software usage in the USPS production environment and a valid Approval Key is what the USPS provided to the Software Vendor or Mailer (who may have developed their own software) as authorization to use in all messages.

ByForConflictQueryRequest Data Structure

Field Description

ByForConflictQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
byForConflictQueryRequest BEGINS				
SubmittingParty	participantID Type complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	SubmittingSoftware complex type	-	Optional	Refer to this complex type in Appendix A
Choice Block BEGINS	-	-	Either userlicensecode/MaildatJobID OR CustomerGroupID/MailingGroupID be provided	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B

ByForConflictQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Required when Mail.XML version is used	
MailingGroupID	Non Negative Integer	-	Required when Mail.XML11.0D is used	
Choice Block ENDS				
byForConflictQueryRequest ENDS				

ByForConflictQueryResponse

This message contains By/For data and is sent upon receiving the request message by customer

Prerequisites

- Customer uses Mail.XML 11.0D spec
- USPS validates request messages
- USPS retrieves the data based on the search criteria provided in the request
- USPS returns response to the Customer

Business Rules

The ByForConflictQueryResponse has two major choice blocks:

1. QueryResults
2. QueryErrors.

- In QueryResults
  - Data will be filtered based on the request criteria
  - The job information will be returned by the USPS *PostalOne!* system, and the Mail.dat Job information will be returned or Mail.XML job information will be returned. The Mail.dat Job ID and User License Code elements represent Mail.dat information and the Customer Group ID and MailingGroupID represent the Mail.XML/Postal Wizard information.
  - The customer will receive one or more ByForConflict blocks
  - The USPS as part of the QueryResults to communicate any issues can return an optional ReturnCode and Description.
- The QueryError Block provides zero or more return codes and descriptions to communicate any issues with the response of the query.

ByForConflictQueryResponse Data Structure

Field Description

ByForConflictQueryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
ByForConflictQueryResponse BEGINS				
LargeTransactionDividerGroupType	LargeTransactionDividerGroupType Attribute	-	Required	Refer this attribute block in Appendix A
TrackingID	String 12	-	Optional Allows user to retrieve the data without resending the request message by using tracking ID in MRRR message	
Choice Block BEGINS			Either QueryResult or QueryError block is returned.	
QueryResult Block BEGINS			Required	
LargeTransactionDividerResult Block BEGINS			Optional	
LargeTransactionDivider BEGINS	-	-	Required. 1 to many allowed	-
TrackingID	String 12		Required	
LargeTransactionDivider ENDS				
LargeTransactionDividerResult Block ENDS				

ByForConflictQueryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
ByForConflict	byForConflict Type complex type		Optional 0 to many allowed	Refer to this complex type in Appendix A
QueryResult Block ENDS				
QueryError Block BEGINS				
ReturnInfo	basicReturnInfo complex type	-	Required	Refer to this complex type in Appendix A
QueryError Block ENDS				
ByForConflictQueryResponse ENDS				

DeliveryResponse

The purpose of this message is to receive an acknowledgment from the customer when they receive a delivery message pushed by USPS.

Prerequisites

- USPS pushes data to the customer's web service end point using delivery messages
- Customers send an acknowledgment using the DeliveryResponse Message

Business Rules:

- USPS expects a "DeliveryResponse" message from all customers who are subscribed to receive "Delivery" messages from the USPS. This response confirms that delivery message has been successfully received by the customer.
- Every Delivery Message will have a unique message identifier (PushMessageID) that will be sent to the customer. Customer are encouraged to return the same PushMessageID in the DeliveryResponse message so that the *PostalOne!* system can reconcile the message delivery information.

DeliveryResponse Data Structure

Field Description

DeliveryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
deliveryResponse BEGINS				

DeliveryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Choice Block BEGINS			Optional. UserLicenseCode/MaildatJobID OR CustomerGroupID/MalingGroupID be provided	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional but required when Mail.XML version is used	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
Push Message ID	String, Length 25	-	Optional	
Delivery Type	String	-	Required	
ReturnInfo	basicReturnInfoType complex type	-	Required	Refer to this complex type in Appendix A
deliveryResponse ENDS				

NotificationResponse

The purpose of this message is to receive an acknowledgment from the customer when they receive a notification message sent by USPS

Prerequisites

- USPS sends a Notification message to the customer's web service end point regarding the availability of data to be pulled by the customer
- Customer send an acknowledgment using the NotificationResponse Message

Business Rules

- **USPS expects a "NotificationResponse" message from all customers who have been subscribed to receive "Notification" messages from the USPS. This response confirms that notification message has been succes**sfully received by the customer.

NotificationResponse Data Structure

Field Description

NotificationResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
NotificationResponse BEGINS				
Choice Block BEGINS			Optional. UserLicenseCode/MaildatJobID OR CustomerGroupID/MalingGroupID be provided	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
PushMessageID	String 25	-	Optional	-
NotificationDate	Date	yyyy-mm-dd	Required	-
ReturnInfo	basicReturnInfo complex type	-	Required	Refer to this complex type in Appendix A
NotificationResponse ENDS				

AddressCorrectionNotification

This is the Notification message to customer for availability of address correction information.

Note: *This message was formerly known as FullServiceAddressCorrectionNotification.*

Prerequisites

Mailer has subscribed to receive AddressCorrectionNotification messages

ACS Data Recipient Determination Process

- Mailer submits eDOC for a Job, Postage Statement is Finalized
- USPS performs By/For Verification
- USPS validate whether Mail Owner MID provided in CPT/MPA/IMR record in eDOC.
- USPS identifies Mail Owner by MID based on Order of Preference
- USPS looks up Data Distribution Profile by Mailer Owner's MID
- USPS verifies whether Data Distribution Profile Exists
- USPS determines Data Recipient CRID based on Profile
- USPS attaches the Data Recipient CRID to the Piece/Piece Range Record

ACS Association and Feedback Process

- USPS receives ACS COA records
- USPS associates ACS COA records with Piece/Piece Range records
- USPS provides ACS Feedback to Data Recipient CRID attached to the Piece/Piece Range record
- USPS pushes the AddressCorrectionNotification message to the Mailer's web service end point

Business Rules

With each AddressCorrectionNotification message, if the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide the

1. SubmittingParty – The *PostalOne!* system for all Notification and Delivery messages will always put a value of 'USPS' in the MaildatUserLicense field.
2. SubmittingSoftware – The *PostalOne!* system for all Notification and Delivery messages will always put:
   a. The value of 'PostalOne' in the SoftwareName element
   b. The value of 'USPS' in the vendor element
   c. The most current version number available, such as 19.0 or 20.0 in the Version element
   d. The *PostalOne!* system will not fill any values in the ApprovalDate and ApprovalKey elements
3. The CRID element is for industry usage and is optional. For the Delivery messages, the *PostalOne!* system will not populate this element.
4. The *PostalOne!* system will provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard to the customer.
5. The *PostalOne!* system will provide the NotificationDate, which is the date when the message is transmitted.

AddressCorrectionNotification Data Structure

Field Description

AddressCorrectionNotification				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
AddressCorrectionNotification BEGINS				
SubmittingParty	participantID Type complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	SubmittingSoftware compelx type	-	Optional	Refer to this complex type in Appendix A
CRID	CRIDType simple type	-	Optional	Refer to this simple type in Appendix B
PushMessageID	String 25	-	Optional	-
Choice Block BEGINS			Optional Choice Block.	
UserLicenseCode	userLicense CodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
Notification Date	**Date**	**yyyy-mm-dd**	**Required**	**-**
FullServiceAvailabilityDate	**Date**	**yyyy-mm-dd**	**Required**	**-**
AddressCorrectionNotification ENDS				

AddressCorrectionDelivery

This is a push message to provide address correction

Prerequisites

Mailer has subscribed to receive AddressCorrectionDelivery messages

ACS Data Recipient Determination Process

- Mailer submits eDOC for a Job, Postage Statement is Finalized
- USPS performs By/For Verification
- USPS validate whether Mail Owner MID provided in CPT/MPA/IMR record in eDOC.
- USPS identifies Mail Owner by MID based on Order of Preference
- USPS looks up Data Distribution Profile by Mailer Owner's MID
- USPS verifies whether Data Distribution Profile Exists
- USPS determines Data Recipient CRID based on Profile
- USPS attaches the Data Recipient CRID to the Piece/Piece Range Record

ACS Association and Feedback Process

- USPS receives ACS COA records
- USPS associates ACS COA records with Piece/Piece Range records
- USPS provides ACS Feedback to Data Recipient CRID attached to the Piece/Piece Range record
- USPS pushes the AddressCorrectionDelivery message to the Mailer's web service end point

Business Rules

With each AddressCorrectionDelivery message, if the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide the

1. OPTIONAL LargeTransactionDividerGroupOptionalType. This type is created to allow breaking up of any large data into multiple transactions, with Feet Ahead concept, where the sender of the large data is providing information about how many total transactions and records to expect and what is the current transaction number among the total expected transactions etc. as part of this one business transaction.
    a. MessageGroupID – It is a Sequence Number and will remain the same until all the transactions are completed, e.g., MessageGroupID is 10, indicating all the transactions with MessageGroupID 10 are one business transaction.
    b. TotalMessageCount - This element tells the receiver that for MessageGroupID of 10 expect a total of X transactions, the X is a positive integer.
    c. MessageSerialNumber – Is a serial number that will always start with 1 for every MessageGroupID, so the receiver always know if this is the first, second, or third transaction from the TotalMessageCount and for the same MessageGroupID.
    d. TransmittedRecordCount – is count of records in the current message transaction.
    e. TotalRecordsAcrossMessages – This is the total expected record count when the receiver of the MessageGroupID has received all messages.
    f. LastMessage – is a Yes or No indicator confirming for the receiver that this transaction is the last transaction of the MessageGroupID e.g., MessageGroupID 10.

2. SubmittingParty – The *PostalOne!* system for all Notification and Delivery messages will always put a value of ‘USPS’ in the MaildatUserLicense field.
3. SubmittingSoftware – The *PostalOne!* system for all Notification and Delivery messages will always put:
    a. The value of ‘PostalOne’ in the SoftwareName element
    b. The value of ‘USPS’ in the vendor element
    c. The most current version number available, such as 19.0 or 20.0 in the Version element
    d. The *PostalOne!* system will not fill any values in the ApprovalDate and ApprovalKey elements
4. The CRID element is for industry usage and is optional. For the Delivery messages, the *PostalOne!* system will not populate this element.
5. The *PostalOne!* system will provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard to the customer.
6. The *PostalOne!* system will send one or more AddressCorrection blocks.
    a. The *PostalOne!* system will provide Move Type to identify the type of move for Individual, Family or Business (indicated by values ‘I’, ‘F’, or ‘B’ respectively).
    b. The COAName element will be used to identify the following:
        i. Individual/Family Move Type = Last Name, First Name, Prefix and Suffix
        ii. Business Move Type = BusinessName will be utilize to provide business name information
    c. The ClassNotificationType element will be used to identify the mail class of the mail piece that generated the ACS COA notification for the following values:
        i. ‘A’ = Package Services (Bound Printed Matter)
        ii. ‘B’ = First-Class
        iii. ‘C’ = Periodicals Initial
        iv. ‘D’ = Standard Mail
        v. ‘F’ = Periodicals Follow-up
    d. The DeliverabilityCodeType element will be used to identify the deliverability status of the mailpiece that generated the ACS COA notification for the following values:
        i. <blank> = COA Match with new address
        ii. ‘K’ = COA Match, no forwarding address
        iii. ‘G’ = COA Match, PO Box Closed, no forwarding address
        iv. ‘W’ = COA Match with a temporary COA
    e. The AddressType element will be used to identify the type of address from which the customer is moving for the following values:
        i. ‘G’ = General Delivery
        ii. ‘H’ = Highway Contract Route
        iii. ‘P’ = Post Office Box
        iv. ‘R’ = Rural Route

v. 'S' = Street Address

AddressCorrectionDelivery Data Structure

Field Description

AddressCorrectionDelivery				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
AddressCorrectionDelivery BEGINS				
LargeTransactionDividerGroup OptionalType	LargeTransactionDividerGroup OptionalType Attribute block	-	Attribute block	Refer this attribute block in Appendix A
SubmittingParty	participantID Type complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	SubmittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
DataRecipient	dataRecipient complex type	-	Optional	Refer to this complex type in Appendix A
PushMessageID	String, Length 25	-	Optional	-
AddressCorrection	FullServiceAddressCorrectionType Complex type	-	Required 1 to many	Refer to this complex type in Appendix A
AddressCorrectionDelivery ENDS				

AddressCorrectionQueryRequest

This is request message to pull COA data

Prerequisites

- Customer uses Mail.XML 11.0D spec
- USPS validates the request messages
- USPS retrieves the data based on the search criteria provided in the request

- USPS returns response to the Customer

Business Rules

With each AddressCorrectionQueryRequest message, the customer needs to provide information for all required blocks. The business rules defined below are not necessarily for required elements.

1. SubmittingParty – The *PostalOne!* system for all data distribution for Full-Service messages, i.e., non-FAST, non-eDoc messages, will only accept a MID or a CRID as a valid id.
2. SubmittingSoftware – The *PostalOne!* system for messages that it receives for the Mail.XML will always require
    a. A valid software name in the SoftwareName element
    b. A valid software vendor company name in the vendor element
    c. The most current version number of the software in the Version element
    d. A valid Approval Date and a valid Approval Key in the ApprovalDate and ApprovalKey elements. The approval date is the date when the USPS approved the software usage in the USPS production environment and a valid Approval Key is what the USPS provided to the Software Vendor or Mailer (who may have developed their own software) as authorization to use in all messages.
3. The OwningParty information can also be provided optionally, if the data is requested by third parties based upon existing profiles or cast of characters messages.
4. The customer shall also provide piece information such as optional piece ranges and job info, as well as date ranges by providing one or more PieceRequest blocks.

AddressCorrectionQueryRequest Data Structure

Field Description

AddressCorrectionQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
AddressCorrectionQueryRequest BEGINS				
SubmittingParty	participantID Type complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
OwningParty	participantID Type complex type	-	Optional	Refer to this complex type in Appendix A

AddressCorrectionQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Piece Request Block BEGINS				
Mail Class	mailClassType simple type	-	Optional	Refer to this simple type in Appendix B
Choice Block BEGINS			Optional	
LowerIMBRange	iMBType simple type	-	Required	Refer to this simple type in Appendix B
UpperIMBRange	iMBType simple type	-	Required	Refer to this simple type in Appendix B
Choice Block ENDS				
Choice Block BEGINS			Optional Choice block. Either UserLicenceCode/MaildatJobID OR CustomerGroupID/MailingGroupID	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				

AddressCorrectionQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Date Range Block BEGINS			Required block	
LowerDateRange	Date	yyyy-mm-dd	Required	This date is a *PostalOne!* date starting in March 2010 to make sure time delay issues when the data is received by *PostalOne!* don't have any impact to the customer querying the data.
UpperDateRange	Date	yyyy-mm-dd	Required	-See Lower Date Range comment above.
Date Range Block ENDS				
Piece Request Block ENDS				
AddressCorrectionQueryRequest ENDS				

AddressCorrectionQueryResponse

This message sends the actual COA data back to the requestor

Prerequisites

- Customer uses Mail.XML 11.0D spec
- USPS validates the request messages
- USPS retrieves the data based on the search criteria provided in the request
- USPS returns response to the Customer

Business Rule

With each AddressCorrectionQueryResponse message, the *PostalOne!* system will provide the

1. Mandatory LargeTransactionDividerGroupType. This type is created to allow breaking up of any large data into multiple transactions, with Feet Ahead concept, where the sender of the

large data is providing information about how many total transactions and records to expect, and what is the current transaction number among the total expected transactions etc. as part of this one business transaction.

a. a. MessageGroupID – It is a Sequence Number and will remain the same until all the transactions are completed, e.g., MessageGroupID is 10 and until the last transaction is received on this business transaction, the MessageGroupID will remain 10 for all transactions, so the receiver knows that all the transactions with MessageGroupID 10 are one business transaction.

b. TotalMessageCount - This element tells the receiver that for MessageGroupID of 10 expect a total of X transactions, the X is a positive integer.

c. MessageSerialNumber – Is a serial number that will always start with 1 for every MessageGroupID, so the receiver always know if this is the 1st, 2nd, 3rd transaction from the TotalMessageCount and for the same MessageGroupID.

d. TransmittedRecordCount – is count of records in the current message transaction.

e. TotalRecordsAcrossMessages – This is the total expected record count when all transactions have been received by the receiver for the MessageGroupID.

f. LastMessage – is an optional Yes or No indicator confirming for the receiver that this transaction is the last transaction of the MessageGroupID e.g., MessageGroupID 10.

The AddressCorrectionQueryResponse has two major choice blocks one is QueryResults and the other is QueryErrors.

- In QueryResults,
  - o The customer will receive zero or more FullServiceAddressCorrection blocks with job information and address correction information.
  - o The customer may also optionally receive Return Codes and Descriptions as part of the Query Results.
- The QueryError Block provides zero or more return codes and descriptions to communicate any issues with the response of the query.

AddressCorrectionQueryResponse Data Structure

Field Description

AddressCorrectionQueryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
AddressCorrectionQueryResponse BEGINS				
LargeTransactionDividerGroupType	LargeTransactionDividerGroupType attribute		Required	Refer this attribute group in Appendix A

AddressCorrectionQueryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
TrackingID	String 12	-	Optional Required when the block data is provided	
Choice Block BEGINS			Either QueryResults block or QueryError block will be returned	
QueryResult Block BEGINS				
LargeTransactionDividerResult	LargeTransactionDividerResult attribute	-	Optional	Refer this attribute block in Appendix A
FullServiceAddressCorrection	FullServiceAddressCorrectionType Complex Type	-	Optional 0 to many	Refer to this complex type in Appendix A
Query Result Block ENDS				
Query Error Block BEGINS				
ReturnInfo	basicReturnInfo complex type	-	Required	Refer to this complex type in Appendix A
Query Error Block ENDS				
Choice Block ENDS				
AddressCorrectionQueryResponse ENDS				

ContainerVisibilityNotification

> ***Note:*** *Starting with Mail.XML 11.0, the Postal Service will share processing scan information of IMcb through Full Service at induction. Container Scan for IMcb in Non-Full Service will be shared during induction through the eInduction process. IMtb tray scan information will also be shared*

with customers for Full Service (Description of the scan type will be clarified in future publication).

This is the Notification message to customer providing them information on their container.

Prerequisites

- Mailer has subscribed to receive ContainerVisibilityNotification messages
- Mailer submits Mail.dat jobs or Mail XML eDoc messages to *PostalOne!*
- USPS validates Mail.XML messages
- USPS processes the Mail.dat jobs or Mail.XML messages, generates postage, and reports
- Postage statement finalization exposes data to SASP
- USPS interprets eDoc information, and performs Full-Service verification
- USPS determines Data Recipients
- USPS sends Mailer Feedback to *PostalOne!*
- USPS Checks the availability of data to the Mailer
- USPS pushes the notification message to the Mailer's web service end point

Business Rules

With each ContainerVisibilityNotification message, if the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide a

- Date of notification

AND

- Provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML to the customer, so the customer can use the Job information in his/her Query request.

ContainerVisibilityNotification Data Structure

Field Description

ContainerVisibilityNotification				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
ContainerVisibilityNotification BEGINS				
SubmittingParty	participantID Type complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
PushMessageID	String 25	-	Optional	-

ContainerVisibilityNotification				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Choice Block BEGINS			Optional	
UserLicenseCode	userLicense CodeType simple type		Required when Mail.dat is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when Mail.dat is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional when Mail.XML is used	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
NotificationDate	Date	yyyy-mm-dd	Required	-
AvailabilityDate	Date	yyyy-mm-dd	Required	-
ContainerVisibilityNotification ENDS				

ContainerVisibilityDelivery

Note: *Starting with Mail.XML 11.0, the Postal Service will share processing scan information of IMcb through Full Service at induction. Container Scan for IMcb in Non-Full Service will be shared during induction through the eInduction process. IMtb tray scan information will also be shared with customers for Full Service (Description of the scan type will be clarified in future publication).*

This is a push message to provide container visibility

Prerequisites

- Mailer has subscribed to receive ContainerVisibilityDelivery messages
- Mailer submits Mail.dat jobs or Mail XML eDoc messages to *PostalOne!*
- USPS validates Mail.XML messages
- USPS processes the Mail.dat jobs or Mail.XML messages, generates postage, and reports
- Postage statement finalization exposes data to SASP
- USPS interprets eDoc information, and performs Full-Service verification
- USPS determines Data Recipients and Start-the-Clock

- USPS sends Mailer Feedback to *PostalOne!*
- USPS checks the availability of data to the Mailer
- USPS pushes the Container Visibility message to the Mailer's web service end point

Business Rules

With each ContainerVisibilityDelivery message, if the customer has registered with the PostalOne! system to receive the message, the *PostalOne!* system will provide the

- OPTIONAL LargeTransactionDividerGroupOptionalType. This type is created to allow breaking up of any large data into multiple transactions, with Feet Ahead concept, where the sender of the large data is providing information about how many total transactions, records to expect, and what is the current transaction number among the total expected transactions etc. as part of this one business transaction.
  - MessageGroupID – It is a Sequence Number and will remain the same until all the transactions are completed, e.g., MessageGroupID is 10 and until the last transaction is received on this business transaction, the MessageGroupID will remain 10 for all transactions, so the receiver knows that all the transactions with MessageGroupID 10 are one business transaction.
  - TotalMessageCount - This element tells the receiver that for MessageGroupID of 10 expect a total of X transactions, the X is a positive integer.
  - MessageSerialNumber – Is a serial number that will always start with 1 for every MessageGroupID, so the receiver always know if this is the 1st, 2nd, 3rd transaction from the TotalMessageCount and for the same MessageGroupID.
  - TransmittedRecordCount – is a count of records in the current message transaction.
  - TotalRecordsAcrossMessages – This is the total expected record count when the receiver for the MessageGroupID has received all transactions.
  - LastMessage – is a Yes or No indicator confirming for the receiver that this transaction is the last transaction of the MessageGroupID. For example: MessageGroupID 10.
- SubmittingParty – The *PostalOne!* system for all Notification and Delivery messages will always put a value of 'USPS' in the MaildatUserLicense field.
- SubmittingSoftware – The *PostalOne!* system for all Notification and Delivery messages will always put
  - The value of 'PostalOne' in the SoftwareName element
  - The value of 'USPS' in the vendor element
  - The most current version number available, such as 19.0 or 20.0 in the Version element
  - The *PostalOne!* system will not fill any values in the ApprovalDate and ApprovalKey elements
- The CRID element is for industry usage and is optional. For the Delivery messages, the *PostalOne!* system will not populate this element.
- The *PostalOne!* system will provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML to the customer.
- The *PostalOne!* system will send one or more ContainerVisibility blocks.
- Electronic Shipping info Received - USPS received electronic Container or Tray information
- Entered at USPS - The Container was Entered at USPS <Facility Name>*

- Enroute Arrive - The Container or Tray arrived at <Facility Name>*
- Enroute - The Tray was processed at <Facility Name>*
- Enroute Departure - The Tray departed from <Facility Name>*
- *Name of USPS facility or airport where scan occurred

ContainerVisibilityDelivery Data Structure

Field Description

ContainerVisibilityDelivery				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
ContainerVisibilityDelivery BEGINS				
LargeTransactionDividerGroup OptionalType	LargeTransactionDividerGroup OptionalType Attribute	-	Required	Refer this attribute block in Appendix A
SubmittingParty	participantID Type complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	SubmittingSoftware complex type	-	Optional	Refer to this complex type in Appendix A
DataRecipient	dataRecipient complex type	-	Optional	Refer to this complex type in Appendix A
PushMessageID	String 25	-	Optional	-
ContainerVisibilityEntry	ContainerVisibilityEntry complex type	-	Required 1 to many	Refer to this complex type in Appendix A
ContainerVisibilityDelivery ENDS				

ContainerVisibilityQueryRequest

This message is sent by customer to pull container visibility data

Prerequisites

- Customer uses Mail.XML 11.0D spec
- USPS validates the request messages
- USPS retrieves the data based on the search criteria provided in the request
- USPS returns response to the Customer

Business Rules

With each ContainerVisibilityQueryRequest message, the customer needs to provide information for all required blocks.

1. SubmittingParty – The *PostalOne!* system for all data distribution for Full-Service messages, i.e., non-FAST, non-eDoc messages, will only accept a MID or a CRID as a valid id.
2. SubmittingSoftware – The *PostalOne!* system for messages that it receives for the Mail.XML will always require
    a. A valid software name in the SoftwareName element
    b. A valid software vendor company name in the vendor element
    c. The most current version number of the software in the Version element
    d. A valid Approval Date and a valid Approval Key in the ApprovalDate and ApprovalKey elements. The approval date is the date when the USPS approved the software usage in the USPS production environment and a valid Approval Key is what the USPS provided to the Software Vendor or Mailer (who may have developed their own software) as authorization to use in all messages.
3. The job information can also be provided optionally, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML by the customer.
4. The FAST appointment ID or the CSA Dispatch ID, created by the FAST system, can be provided optionally in the ConsigneeApptID or CSADispatchNumber elements respectively.
5. The customer can also optionally provide the 21-digit Pallet IMCB barcodes in the UniqueContainerBarcode element tied to the Mail.XML job information or Mail.dat Job information and can optionally provide the Locale Key of the Induction facility in the USPSInductionFacilityLocaleKey element.
6. The customer can also optionally provide the Mail.XML Container ID in the CustomerContainerID element tied to the Mail.XML job information or can optionally provide the Mail.dat ContainerID in the ContainerID element.

ContainerVisibilityQueryRequest Data Structure

Field Description

ContainerVisibilityQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
ContainerVisibilityQueryRequest BEGINS				
SubmittingParty	participantID Type complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwarType complex type	-	Optional	Refer to this complex type in Appendix A
Choice Block BEGINS			Optional	

ContainerVisibilityQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
UserLicenseCode	userLicense CodeType simple type		Required when Mail.dat is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when Mail.dat is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Required when Mail.dat is used	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
Choice Block BEGINS			Optional	
ConsigneeApptID	Numeric String,12	-	Optional	-
CSAID		-	Optional	
Choice Block ENDS				
IMcb	IMcbType simple type	-	Optional 0 to many	Refer to this simple type in Appendix B
USPSInductionFacilitylocaleKey	localeKeyType simple type	-	Optional	Refer to this simple type in Appendix B
ContainerVisibilityQueryRequest ENDS				

ContainerVisibilityQueryResponse

This message contains container visibility data and is sent upon receiving the request message

Prerequisites

- Customer uses Mail.XML 11.0D spec
- USPS validates the request messages
- USPS retrieves the data based on the search criteria provided in the request
- USPS returns response to the Customer

Business Rules

With each ContainerVisibilityQueryResponse message the *PostalOne!* system will provide the

1. Mandatory LargeTransactionDividerGroupType. This type is created to allow breaking up of any large data into multiple transactions, with Feet Ahead concept, where the sender of the large data is providing information about how many total transactions and records to expect, and what is the current transaction number among the total expected transactions etc. as part of this one business transaction.

   a. MessageGroupID – It is a Sequence Number and will remain the same until all the transactions are completed, e.g., MessageGroupID is 10 and until the last transaction is received on this business transaction, the MessageGroupID will remain 10 for all transactions, so the receiver knows that all the transactions with MessageGroupID 10 are one business transaction.

   b. TotalMessageCount - This element tells the receiver that for MessageGroupID of 10 expect a total of X transactions, the X is a positive integer.

   c. . MessageSerialNumber – Is a serial number that will always start with 1 for every MessageGroupID, so the receiver always know if this is the 1st, 2nd, 3rd transaction from the TotalMessageCount and for the same MessageGroupID.

   d. TransmittedRecordCount – is count of records in the current message transaction.

   e. TotalRecordsAcrossMessages – This is the total expected record count when the receiver of the MessageGroupID has received all messages.

   f. LastMessage – is an optional Yes or No indicator confirming for the receiver that this transaction is the last transaction of the MessageGroupID e.g., MessageGroupID 10.

The ContainerVisibilityQueryResponse has two major choice blocks one is QueryResults and the other is QueryErrors.

In QueryResults,

- The job information will be returned by the USPS PostalOne! system, and the Mail.dat Job information will be returned or Mail.XML job information will be returned. The Mail.dat Job ID and User License Code elements represent Mail.dat information and the Customer Group ID and MailingGroupID represent the Mail.XML.
- The customer will receive zero or more ContainerVisibilityEntry blocks with container unload and scan information. The following Scan Types will be provided:
  - o - Electronic Shipping info received - USPS received electronic Container or Tray information
  - o - Entered at USPS - The Container was Entered at USPS <Facility Name>*
  - o - Enroute Arrive - The Container or Tray arrived at <Facility Name>*
  - o Enroute - The Tray was processed at <Facility Name>*
  - o - Enroute Departure - The Tray departed from <Facility Name>*
  - o - Out for Delivery - The Tray was sent out for delivery
  - o *Name of USPS facility or Airport where scan occurred
  - o The customer may also optionally receive Return Codes and Descriptions as part of the Query Results.
- The QueryError Block provides zero or more return codes and descriptions to communicate any issues with the response of the query.

ContainerVisibilityQueryResponse Data Structure

Field Description

ContainerVisibilityQueryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
ContainerVisibilityQueryResponse BEGINS				
LargeTransactionDividerGroupType	LargeTransactionDividerGroupType attribute	-	Required	Refer this attribute block in Appendix A
Tracking ID	String 12	-	Optional. Allows the user to retrieve the date without requerying	-
Choice Block BEGINS			Either QueryResult or QueryError will be returned	
QueryResult block BEGINS			1 to many allowed	
LargeTransactionDividerResult	LargeTransactionDividerResult attribute	-	Optional	Refer this attribute block in Appendix A
ContainerVisibilityEntry	ContainerVisibilityEntry complex type	-	Optional 0 to many allowed	Refer to this complex type in Appendix A
ReturnInfo	basicReturnInfo complex type	-	Optional	Refer to this complex type in Appendix A
Query Result Block ENDS				
Query Error Block BEGINS				
ReturnInfo	basicReturnInfo complex type	-	Required	Refer to this complex type in Appendix A

ContainerVisibilityQueryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Query Error Block ENDS				
ContainerVisibilityQueryResponse ENDS				

DataQualityVerificationReportNotification

This is notification message sent by USPS to customer

Prerequisites

Mailer has subscribed to receive DataQualityVerificationReportNotification messages

- eDoc Data Quality Data Recipient Determination Process
  - Mailer submits eDOC for a Job,
  - Postage Statement is Finalized
  - USPS performs By/For Verification
  - USPS validate whether Mail Owner MID provided in CPT/MPA/IMR record in eDOC.
  - USPS identifies Mail Owner by MID based on Order of Preference
  - USPS identifies Mail Preparer
- eDoc Data Quality Logging and Feedback Process
  - USPS performs verifications on eDoc submitted
  - USPS associates error records with eDoc containers, handling units, piece or piece range records
- USPS pushes the *DataQualityVerificationReportNotification* message to the Mailer's web service end point

Business Rules:

If the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide a date of notification and will provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard to the customer, and report type

DataQualityVerificationReportNotification Data Structure

Field Description

DataQualityVerificationReportNotification				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
DataQualityVerificationReportNotification BEGINS				

DataQualityVerificationReportNotification				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
SubmittingParty	participantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
PushMessageID	String 25	-	Optional	-
Choice Block BEGINS			Required.	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
eDocSenderCRID	CRIDType simple type		Optional	Refer to this simple type in Appendix B
VerificationReport Type	verificationReportType simple type	-	Required 1 to many allowed	Refer to this simple type in Appendix B
AvailabilityDate	Date	yyyy-mm-dd	Required	-
NotificationDate	Date	yyyy-mm-dd	Required	
DataQualityVerificationReportNotification ENDS				

DataQualityVerificationReportDelivery

This message is pushed by USPS and contains data quality verification report

Prerequisites

Mailer has subscribed to receive *DataQualityVerificationReportDelivery* messages

eDoc Data Quality Data Recipient Determination Process

- Mailer submits eDOC for a Job,
- Postage Statement is Finalized
- USPS performs By/For Verification
- USPS validate whether Mail Owner MID provided in CPT/MPA/IMR record in eDOC.
- USPS identifies Mail Owner by MID based on Order of Preference
- USPS identifies Mail Preparer

eDoc Data Quality Logging and Feedback Process

  - o USPS performs verifications on eDoc submitted
  - o USPS associates error records with eDoc containers, handling units, piece or piece range records

- USPS pushes the DataQualityVerificationReportDelivery message to the Mailer's web service end point

Business Rules:

If the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide a date of notification and will provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard to the customer, report type, and details about the error

DataQualityVerificationReportDelivery Data Structure

Field Description

DataQualityVerificationReportDelivery				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
DataQualityVerification ReportDelivery BEGINS				
LargeTransaction DividerGroupOptional Type	LargeTransaction DividerGroupOptional Type Attribute	-	Required	Refer this attribute block in Appendix A
SubmittingParty	participantIDType complex type		Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type		Optional	Refer to this complex type in Appendix A
DataRecipient	dataRecipient complex type	-	Optional	Refer to this complex type in Appendix A

DataQualityVerificationReportDelivery				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
Choice Block BEGINS				
UserLicenseCode	userLicenseCode Type simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
eDocSenderCRID	CRIDType simple type		Required	Refer to this simple type in Appendix B
PushMessageID	String 25	-	Optional	-
TotalJobContainerCount	nonnegative Integer		Optional	
TotalJobHandlingUnitCount	nonnegative Integer		Optional	
TotalJobPieceCount	nonnegative Integer		Optional	
TotalJobContainerHUErrorCount	nonnegative Integer		Optional	
TotalJobPieceErrorCount	nonnegative Integer		Optional	
TotalJobContainerHUWarningCount	nonNegativeInteger			
TotalJobPieceWarningCount	nonNegativeInteger			
Sequence BEGINS			0 to many	
PreparerCRID	CRIDType simple type		Required	Refer to this simple type in Appendix B

DataQualityVerificationReportDelivery				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
TotalPreparerContainerCount	nonnegative Integer		Required	
TotalPreparerHandlingUnitCount	xs:nonNegativeInteger		Required	
TotalPreparerPieceCount	nonnegative Integer		Required	
TotalPreparerContainerHUErrorCount	nonnegative Integer		Required	
TotalPreparerContainerHUWarningCount	nonNegativeInteger			
TotalPreparerPieceWarningCount	nonNegativeInteger			
TotalPreparerPieceErrorCount	nonnegative Integer		Required	
Sequence ENDS				
Sequence BEGINS			Optional 0 to many allowed	
OwnerCRID	CRIDType simple type		Required	Refer to this simple type in Appendix B
TotalOwnerContainerCount	nonnegative Integer		Required	
TotalOwnerHandlingUnitCount	nonnegative Integer		Required	
TotalOwnerPieceCount	nonnegative Integer		Required	
TotalOwnerContainerHUErrorCount	nonnegative Integer		Required	
TotalOwnerPieceErrorCount	nonnegative Integer		Required	
TotalOwnerContainerHUWarningCount	nonnegative Integer			
TotalOwnerPieceWarningCount	nonnegative Integer			
Sequence ENDS			-	

DataQualityVerificationReportDelivery				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
Sequence BEGINS			Required 1 to many allowed	
VerificationReportType	verificationReport Type simple type	-	Required	Refer to this simple type in Appendix B
DataQuality ContainerInfo	dqrContainerInfo Type complex type		Required 1 to many allowed	Refer to this complex type in Appendix A
Sequence ENDS				
DataQualityVerification ReportDelivery ENDS				

DataQualityVerificationReportQueryRequest

Customers send this message to USPS for the pull data quality verification error report.

Prerequisites

- Mailer submits Mail.dat jobs, Mail XML or Postal Wizard eDoc to *PostalOne!*
- USPS validates Mail.XML messages
- USPS processes the Mail.dat jobs, Mail.XML messages or Postal Wizard
- USPS generates postage and reports
- Postage statement has been finalized by USPS
- USPS interprets eDoc information, and performs Full-Service verification
- USPS checks the availability of eDoc data quality information to the Mailer
- USPS responds to the DataQualityVerificationReportQueryRequest message to the Mailer's web service end point.

Business Rules

The customer must provide job information, submitting party, submitting software, facility id, and the report type to retrieve the data quality verification report.

DataQualityVerificationReportQueryRequest Data Structure

Field Description

DataQualityVerificationReportQueryRequest				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**

DataQualityVerificationReportQueryRequest				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
DataQualityVerificationReportQueryRequest BEGINS				
SubmittingParty	participantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
Choice Block BEGINS			Required	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
eDocSenderCRID	CRIDType simple type		Optional	Refer to this simple type in Appendix B
VerificationReportType	mailxml_dd:verificationReportType	-	Required 1 to many	Refer to this simple type in Appendix B
VerificationErrorType	mailxml_dd:verificationErrorType	-		
VerificationWarningType	mailxml_dd:verificationWarningType	-		

DataQualityVerificationReportQueryRequest				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
DataQualityVerificationReportQueryRequest ENDS				

DataQualityVerificationReportQueryResponse

This message is sent by USPS in response to query request

Prerequisites

- Mailer sends DataQualityVerificationReportQueryRequest message to USPS
- USPS validates Mail.XML message and processes request

Error/Return Code

Refer to Section 2 for the list of Error/Return code tied to this message and all shared error/return codes

Business Rules

If the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide a date of notification and will provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard to the customer, report type, and details about the error

DataQualityVerificationReportQueryResponse Data Structure

Field Description

DataQualityVerificationReportQueryResponse				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
DataQualityVerificationReportQueryResponse BEGINS				
LargeTransactionDividerGroupOptionalType	LargeTransactionDividerGroupOptional Type Attribute block	-	Required	Refer this attribute block in Appendix A

DataQualityVerificationReportQueryResponse				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
TrackingID	String 12	-	Optional Allows the user to retrieve the data without requerying	-
Choice Block BEGINS			Either QueryResult or QueryError block is returned	
QueryResult Block BEGINS			Required	
SubmittingParty	participantIDType complex type		Required	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type		Required	Refer to this complex type in Appendix A
Choice Block BEGINS			Required Choice Block	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				

DataQualityVerificationReportQueryResponse				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
LargeTransactionDividerResult	LargeTransaction DividerResult attribute	-	Optional	Refer this attribute block in Appendix A
eDocSenderCRID	CRIDType simple type		Required	Refer to this simple type in Appendix B
TotalJobContainerCount	nonnegative Integer		Optional	
TotalJobHandlingUnitCount	nonnegative Integer		Optional	
TotalJobPieceCount	nonnegative Integer		Optional	
TotalJobContainerHU ErrorCount	nonnegative Integer		Optional	
TotalJobPieceError Count	nonnegative Integer		Optional	
Sequence BEGINS			Optional 0 to many allowed	
PreparerCRID	CRIDType simple type		Required	Refer to this simple type in Appendix B
TotalPreparerContainer Count	nonnegative Integer		Required	
TotalPreparerHandling UnitCount	xs:nonNegativeInteger		Required	
TotalPreparerPiece Count	nonnegative Integer		Required	
TotalPreparerContainer HUErrorCount	nonnegative Integer		Required	
TotalPreparerPieceError Count	nonnegative Integer		Required	
Sequence ENDS				
Sequence BEGINS			Optional 0 to many allowed	
OwnerCRID	CRIDType simple type		Required	Refer to this simple type in Appendix B

DataQualityVerificationReportQueryResponse				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
TotalOwnerContainer Count	nonnegative Integer		Required	
TotalOwnerHandlingUnit Count	nonnegative Integer		Required	
TotalOwnerPieceCount	nonnegative Integer		Required	
TotalOwnerContainerHU ErrorCount	nonnegative Integer		Required	
TotalOwnerPieceError Count	nonnegative Integer		Required	
Sequence ENDS			Required	
Sequence BEGINS			Required 1 to many	
VerificationReportType	verificationReport Type	-	Required	Refer to this simple type in Appendix B
DQRContainerInfo	dqrContainerInfoT ype complex type		Required	Refer to this complex type in Appendix A
Sequence ENDS				
QueryResult Block ENDS				
QueryError Block BEGINS				
ReturnInfo	basicReturnInfo complex type	-	Required	Refer to this complex type in Appendix A
QueryError block ENDS				
Choice Block ENDS				
DataQualityVerification ReportQueryResponse ENDS				

NixieDetailNotification

This is the Notification message to customer for Nixie information.

Prerequisites

Mailer has subscribed to receive NixieDetailNotification messages

ACS Data Recipient Determination Process

- Mailer submits eDOC for a Job, Postage Statement is Finalized
- USPS performs By/For Verification

- USPS validate whether Mail Owner MID provided in CPT/MPA/IMR record in eDOC.
- USPS identifies Mail Owner by MID based on Order of Preference
- USPS looks up Data Distribution Profile by Mailer Owner’s MID
- USPS verifies whether Data Distribution Profile Exists
- USPS determines Data Recipient CRID based on Profile
- USPS attaches the Data Recipient CRID to the Piece/Piece Range Record

ACS Association and Feedback Process

- USPS receives ACS and Nixie records
- USPS associates ACS and Nixie records with Piece/Piece Range records
- USPS provides ACS Feedback to Data Recipient CRID attached to the Piece/Piece Range record
- USPS pushes the NixieDetailNotification message to the Mailer’s web service end point

Business Rules

With each NixieDetailNotification message, if the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide the

1. SubmittingParty – The *PostalOne!* system for all Notification and Delivery messages will always put a value of ‘USPS’ in the MaildatUserLicense field.
2. SubmittingSoftware – The *PostalOne!* system for all Notification and Delivery messages will always put
    a. The value of ‘PostalOne’ in the SoftwareName element
    b. The value of ‘USPS’ in the vendor element
    c. The most current version number available, such as 19.0 or 20.0 in the Version element
    d. The *PostalOne!* system will not fill any values in the ApprovalDate and ApprovalKey elements
3. The CRID element is for industry usage and is optional. For the Delivery messages, the *PostalOne!* system will not populate this element.
4. The *PostalOne!* system will provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard to the customer.
5. The *PostalOne!* system will provide the NotificationDate, which is the date when the message is transmitted.

NixieDetailNotification Data Structure

Field Description

NixieDetailNotification				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
NixieDetailNotification BEGINS				
SubmittingParty	participantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
CRID	CRIDType simple type	-	Optional	Refer this complex type in Appendix B
PushMessageID	String 25	-	Optional	-
Choice Block BEGINS			Optional. Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID be provided	
UserLicenseCode	userLicenseCode Type simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
AvailabilityDate	Date	yyyy-mm-dd	Required	-
Notification Date	Date	yyyy-mm-dd	Required	-
NixieDetailNotification ENDS				

NixieDetailDelivery

This is a push message to provide Nixie information

Prerequisites

Mailer has subscribed to receive NixieDetailDelivery messages

ACS Data Recipient Determination Process

- Mailer submits eDOC for a Job, Postage Statement is Finalized
- USPS performs By/For Verification
- USPS validate whether Mail Owner MID provided in CPT/MPA/IMR record in eDOC.
- USPS identifies Mail Owner by MID based on Order of Preference
- USPS looks up Data Distribution Profile by Mailer Owner's MID
- USPS verifies whether Data Distribution Profile Exists
- USPS determines Data Recipient CRID based on Profile
- USPS attaches the Data Recipient CRID to the Piece/Piece Range Record

ACS Association and Feedback Process

- USPS receives ACS and Nixie records
- USPS associates ACS and Nixie records with Piece/Piece Range records
- USPS provides ACS Feedback to Data Recipient CRID attached to the Piece/Piece Range record
- USPS pushes the NixieDetailDelivery message to the Mailer's web service end point

Business Rule

With each NixieDetailDelivery message, if the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide the

1. OPTIONAL LargeTransactionDividerGroupOptionalType. This type is created to allow breaking up of any large data into multiple transactions, with Feet Ahead concept, where the sender of the large data is providing information about how many total transactions and records to expect and what is the current transaction number among the total expected transactions etc. as part of this one business transaction.

   a. MessageGroupID – It is a Sequence Number and will remain the same until all the transactions are completed, e.g., MessageGroupID is 10 and until the last transaction is received on this business transaction, the MessageGroupID will remain 10 for all transactions, so the receiver knows that all the transactions with MessageGroupID 10 are one business transaction.

   b. TotalMessageCount - This element tells the receiver that for MessageGroupID of 10 expect a total of X transactions, the X is a positive integer.

   c. MessageSerialNumber – Is a serial number that will always start with 1 for every MessageGroupID, so the receiver always know if this is the 1st, 2nd, 3rd transaction from the TotalMessageCount and for the same MessageGroupID.

   d. TransmittedRecordCount – is count of records in the current message transaction.

   e. TotalRecordsAcrossMessages – This is the total expected record count when the receiver of the MessageGroupID has received all messages.

f. LastMessage – is a Yes or No indicator confirming for the receiver that this transaction is the last transaction of the MessageGroupID e.g., MessageGroupID 10.

2. SubmittingParty – The *PostalOne!* system for all Notification and Delivery messages will always put a value of 'USPS' in the MaildatUserLicense field.
3. SubmittingSoftware – The *PostalOne!* system for all Notification and Delivery messages will always put
   a. The value of 'PostalOne' in the SoftwareName element
   b. The value of 'USPS' in the vendor element
   c. The most current version number available, such as 19.0 or 20.0 in the Version element
   d. The *PostalOne!* system will not fill any values in the ApprovalDate and ApprovalKey elements
4. The CRID element is for industry usage and is optional. For the Delivery messages, the *PostalOne!* system will not populate this element.
5. The *PostalOne!* system will provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard to the customer.
6. The *PostalOne!* system will send one or more FullServiceNixieDetail blocks.
   a. The ActionCode element will be used to identify the action taken for the returned mail for the following values:
      i. 'F' = Forwarded
      ii. 'W' = Wasted
      iii. 'R' = Returned
      iv. 'U' = Unable to determine action
   b. b. The ClassNotificationType element will be used to identify the mail class of the mail piece that generated the ACS COA notification for the following values:
      i. 'A' = Package Services (Bound Printed Matter)
      ii. 'B' = First-Class
      iii. 'C' = Periodicals Initial
      iv. 'D' = Standard Mail
      v. 'F' = Periodicals Follow-up
   c. Reason Code - Reason mail element will be used to identify the reason for the returned mail for the following values:
      i. 'A' = Attempted, not known
      ii. 'B' = Returned for better address
      iii. 'D' = Outside delivery limits
      iv. 'E' = In dispute
      v. 'I' = Insufficient Address
      vi. 'L' = Illegible
      vii. 'M' = No mail receptacle

viii. 'N' = No such number

ix. 'P' = Deceased

x. 'Q' = Not deliverable as addressed/unable to forward

xi. 'R' = Refused

xii. 'S' = No such street

xiii. 'U' = Unclaimed

xiv. 'V' = Vacant

xv. 'X' = No such office

NixieDetailDelivery Data Structure

Field Description

NixieDetailDelivery				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
NixieDetailDelivery BEGINS				
LargeTransactionDividerGroup OptionalType	LargeTransactionDividerGroup OptionalType Attribute	-	Required	Refer this attribute block in Appendix A
SubmittingParty	participantID Type complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
DataRecipient	dataRecipient complex type	-	Optional	Refer to this complex type in Appendix A
Push Message ID	String, Length 25	-	Optional	-
FullServiceNixieDetail	FullServiceNixieDetailType Complex type	-	Required 1 to many allowed	Refer to this complex type in Appendix A
NixieDetailDelivery ENDS				

NixieDetailQueryRequest

This is request message to pull Nixie data

Prerequisites

- Customer uses Mail.XML 11.0D spec
- USPS validates the request messages
- USPS retrieves the data based on the search criteria provided in the request
- USPS returns response to the Customer

Business Rules

With each NixieDetailQueryRequest message, the customer needs to provide information for all required blocks. See below the business rule specific to fields.

1. SubmittingParty – The *PostalOne!* system for all data distribution for Full-Service messages, i.e., non-FAST, non-eDoc messages, will only accept a MID or a CRID as a valid id.
2. SubmittingSoftware – The *PostalOne!* system for messages that it receives for the Mail.XML will always require
    a. A valid software name in the SoftwareName element
    b. A valid software vendor company name in the vendor element
    c. The most current version number of the software in the Version element
    d. A valid Approval Date and a valid Approval Key in the ApprovalDate and ApprovalKey elements. The approval date is the date when the USPS approved the software usage in the USPS production environment and a valid Approval Key is what the USPS provided to the Software Vendor or Mailer (who may have developed their own software) as authorization to use in all messages.
3. The RequestorCRID can be used to identify the requestor through the company CRID.
4. The customer can also provide the 6-digit or 9-digit OriginalMailerID.
5. The customer shall also provide one or more PieceRequests with piece ranges, class, date range and job information.

NixieDetailQueryRequest Data Structure

Field Description

NixieDetailQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
NixieDetailQueryRequest BEGINS				
SubmittingParty	participantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingPartyType complex type	-	Optional	Refer to this complex type in Appendix A
Requester CRID	CRIDType simple type	-	Optional	Refer to this simple type in Appendix B

NixieDetailQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Choice Block BEGINS			Optional Block	
OriginalMailerID6	mailerID6Type simple type	-	Required	Refer to this simple type in Appendix B
OriginalMailerID9	mailerID9Type simple type	-	Required	Refer to this simple type in Appendix B
Choice Block ENDS				
Piece Request Block BEGINS			Required 1 to many allowed	
Mail Class	mailClassType simple type	-	Optional	Refer to this simple type in Appendix B
Choice Block BEGINS			Optional Either Lower or Upper IMBRange is required	
LowerIMBRange	iMBType simple type	-	Required	Refer to this simple type in Appendix B
UpperIMBRange	iMBType simple type	-	Required	Refer to this simple type in Appendix B
Choice Block ENDS				
Choice Block BEGINS			Optional Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID be provided	

NixieDetailQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
Date Range Block BEGINS			Required	
LowerDateRange	Date	yyyy-mm-dd	Required	This date is a *PostalOne!* date starting in March 2010 to make sure time delay issues when the data is received by *PostalOne!* Do not have any impact to the customer querying the data.
UpperDateRange	Date	yyyy-mm-dd	Required	-See Lower Date Range comment above.
Date Range Block ENDS				
Piece Request Block ENDS				
NixieDetailQueryRequest ENDS				

NixieDetailQueryResponse

This message sends the actual Nixie data back to the requestor

Prerequisites

- Customer uses Mail.XML 11.0D spec
- USPS validates the request messages

- USPS retrieves the data based on the search criteria provided in the request
- USPS returns response to the Customer

Business Rules

With each NixieDetailQueryResponse message, the *PostalOne!* system will provide the

1. Mandatory LargeTransactionDividerGroupType. This type is created to allow breaking up of any large data into multiple transactions, with Feet Ahead concept, where the sender of the large data is providing information about how many total transactions and records to expect, and what is the current transaction number among the total expected transactions etc. as part of this one business transaction.
    a. MessageGroupID – It is a Sequence Number and will remain the same until all the transactions are completed, e.g., MessageGroupID is 10 and until the last transaction is received on this business transaction, the MessageGroupID will remain 10 for all transactions, so the receiver knows that all the transactions with MessageGroupID 10 are one business transaction.
    b. TotalMessageCount - This element tells the receiver that for MessageGroupID of 10 expect a total of X transactions, the X is a positive integer.
    c. MessageSerialNumber – Is a serial number that will always start with 1 for every MessageGroupID, so the receiver always know if this is the 1st, 2nd, 3rd transaction from the TotalMessageCount and for the same MessageGroupID.
    d. TransmittedRecordCount – is count of records in the current message transaction.
    e. TotalRecordsAcrossMessages – This is the total expected record count when the receiver of the MessageGroupID has received all messages LastMessage – is an optional Yes or No indicator confirming for the receiver that this transaction is the last transaction of the MessageGroupID e.g., MessageGroupID 10.

The NixieDetailResponse has two major choice blocks one is QueryResults and the other is QueryErrors.

- In QueryResults,

The customer will receive zero or more FullServiceNixieDetail blocks with job info, original MID information and Nixie address information.

The customer may also optionally receive Return Codes and Descriptions as part of the Query Results.

- The QueryError Block provides zero or more return codes and descriptions to communicate any issues with the response of the query.

NixieDetailQueryResponse Data Structure

Field Description

NixieDetailQueryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
NixieDetailQueryResponse BEGINS				
LargeTransactionDividerGroupType	LargeTransactionDividerGroupType Attribute	-	Required	Refer this attribute block in Appendix A

NixieDetailQueryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
TrackingID	String 12	-	Optional. Allows the user to retrieve the data without requerying it again	-
Choice Block BEGINS			Required Either QueryResult or QueryError will be returned	
Query Result Block BEGINS				
LargeTransactionDividerResult	LargeTransactionDividerResult Attribute	-	Optional	Refer this attribute block in Appendix A
FullServiceNixieDetail	fullServiceNixieDetailType complex type	-	Optional. 0 to many allowed	Refer to this complex type in Appendix A
ReturnInfo	basicReturnInfo complex type	-	Optional	Refer to this complex type in Appendix A
Query Result Block ENDS				
Query Error Block BEGINS				
ReturnInfo	basicReturnInfo complex type	-	Required	Refer to this complex type in Appendix A
Query Error Block ENDS				
NixieDetailQueryResponse ENDS				

StartTheClockNotification

The purpose of this message is to provide notification to customers with the StartTheClock information

Prerequisites

- Mailer has subscribed to receive StartTheClock Notification messages
- Mailer submits Mail.dat jobs or Mail XML eDoc messages to *PostalOne!*
- USPS validates Mail.XML messages
- USPS processes the Mail.dat jobs or Mail.XML messages, generates postage, and reports
- Postage statement finalization exposes data to SASP
- USPS interprets eDoc information, and performs Full-Service verification

- USPS determines Data Recipients and Start-the-Clock
- USPS sends Mailer Feedback to *PostalOne!*
- USPS Checks the availability of Start the Clock data to the Mailer
- USPS pushes the notification message to the Mailer's web service end point

Business Rules

- If the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide a date of notification and will provide the job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard to the customer, so the customer can use the Job information in his/her Query request.

StartTheClockNotification Data Structure

Field Description

StartTheClockNotification				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
StartTheClockNotification BEGINS				
SubmittingParty	participantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
DataRecipient	dataRecipient complex type	-	Optional	Refer to this complex type in Appendix A
PushMessageID	String 25	-	Optional	-
Choice Block BEGINS			Optional Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID be provided	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B

StartTheClockNotification				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
FullServiceAvailabilityDate	Date	yyyy-mm-dd	Required	
Notification Date	Date	yyyy-mm-dd	Required	
StartTheClockNotification ENDS				

StartTheClockDelivery

The purpose of this message is to deliver the StartTheClock data to recipient server i.e. customer's server

Prerequisites

- Mailer has subscribed to receive StartTheClock Delivery messages
- Mailer submits Mail.dat jobs or Mail XML eDoc messages to *PostalOne!*
- USPS validates Mail.XML messages
- USPS processes the Mail.dat jobs or Mail.XML messages, generates postage, and reports
- Postage statement finalization exposes data to SASP
- USPS interprets eDoc information, and performs Full-Service verification
- USPS determines Data Recipients and Start-the-Clock
- USPS sends Mailer Feedback to *PostalOne!*
- USPS Checks the availability of Start the Clock data to the Mailer
- USPS pushes Start The Clock delivery data to the Mailer's web service end point

Business Rules

If the customer has registered with the *PostalOne!* system to receive the message, the *PostalOne!* system will provide the:

- Job information, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard to the customer.
- Zero or more StartTheClockBMEUBlock blocks for the job
- Zero or more StartTheClockDropShipBlock blocks for the job
- Zero or more StartTheClockPlantLoadBlock blocks for the job
- In addition, optional return Info block in case of issues or no data found. All return codes and their descriptions for the Response messages will be made available by the USPS for implementation based upon the schedule made available in the External Timeline for IM implementation document.

StartTheClockDelivery Data Structure

Field Description

StartTheClockDelivery

Field	Format	Acceptable Values	Business Rules	Comments
StartTheClockDelivery BEGINS				
SubmittingParty	participantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
DataRecipient	dataRecipient complex type	-	Optional	Refer to this complex type in Appendix A
Push Message ID	String 25	-	Optional	-
ClockStarted	clockStartedType complex type	-	Required 1 to many allowed	Refer to this complex type in Appendix A
StartTheClockDelivery ENDS				

StartTheClockQueryRequest

This message is sent by customer to pull StartTheClock data

Prerequisites

- Customer uses Mail.XML 11.0D spec
- USPS validates the request messages
- USPS retrieves the data based on the search criteria provided in the request
- USPS returns response to the Customer

Business Rules

With each StartTheClockQueryRequest message, the customer needs to provide information for all required blocks. Business rule for some of the fields are given below – some of them are optional fields.

1. SubmittingParty – The *PostalOne!* system for all data distribution for Full-Service messages, i.e., non-FAST, non-eDoc messages, will only accept a MID or a CRID as a valid id.
2. SubmittingSoftware – The *PostalOne!* system for messages that it receives from the customer for the Mail.XML will always require
   a. A valid software name in the SoftwareName element
   b. A valid software vendor company name in the vendor element
   c. The most current version number of the software in the Version element
   d. A valid Approval Date and a valid Approval Key in the ApprovalDate and ApprovalKey elements. The approval date is the date when the USPS approved the software usage in the USPS production environment and a valid Approval Key is what the USPS provided to the Software Vendor or Mailer (who may have developed their own software) as authorization to use in all messages.

3. The FAST appointment ID or the CSA Dispatch ID, created by the FAST system, can be provided optionally in the ConsigneeApptID or CSADispatchNumber elements respectively.
4. The job information can also be provided optionally, either the Mail.dat Job ID and User License Code, OR the Customer Group ID and MailingGroupID for Mail.XML/Postal Wizard by the customer.
5. The customer can also optionally provide the Permit information or publication number information to receive the StartTheClock data in PermitNumber, PermitType, and PermitZip4 elements or in the PublicationNumber element.
6. The customer must provide a date range in the LowerDateRange and UpperDateRange elements for all query requests.

StartTheClockQueryRequest Data Structure

Field Description

StartTheClockQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
StartTheClockQueryRequest BEGINS				
SubmittingParty	participantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Required	Refer to this complex type in Appendix A
Choice Block BEGINS			Optional If provided then either ConsigneeApptID or CSADispatchNumber is required	
ConsigneeApptID	String 12	-	Required	-
CSADispatchNumber	String 10	-	Required	-
Choice Block ENDS				

StartTheClockQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Choice Block BEGINS			Optional Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID or CustomerAccount block is required	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
CustomerAccount	permitPublicationDataType complex type	-	Required	Refer to this complex type in Appendix A
Choice Block ENDS				
LowerDateRange	Date	yyyy-mm-dd	Required	This date is a *PostalOne!* date starting in March 2010 to make sure time delay issues when the data is received by *PostalOne!* Do not have any impact to the customer querying the data.

StartTheClockQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
UpperDateRange	Date	yyyy-mm-dd	Required	-See Lower Date Range comment above.
StartTheClockQueryRequest ENDS				

StartTheClockQueryResponse

This message contains StartTheClock data and is sent upon receiving the request message

Prerequisites

- Customer uses Mail.XML 11.0D spec
- USPS validates the request messages
- USPS retrieves the data based on the search criteria provided in the request
- USPS returns response to the Customer

Business Rules

The StartTheClockQuery Response has two major choice blocks one is QueryResults and the other is QueryErrors.

1. In QueryResults,

Data will be filtered based on the request criteria

The job information will be returned by the USPS *PostalOne!* system, and the Mail.dat Job information will be returned or Mail.XML job information will be returned. The Mail.dat Job ID and User License Code elements represent Mail.dat information and the Customer Group ID and MailingGroupID represent the Mail.XML/Postal Wizard information.

The customer will receive one or more ClockStarted blocks. The block includes StartTheClockBMEUBlock, StartTheClockPlantLoadBlock, and StartTheClockDropShipBlock.

The USPS as part of the QueryResults to communicate any issues can return an optional ReturnCode and Description.

2. The QueryError Block provides zero or more return codes and descriptions to communicate any issues with the response of the query.

StartTheClockQueryResponse Data Structure

Field Description

StartTheClockQueryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
StartTheClockQueryResponse BEGINS				
LargeTransactionDividerGroupType	LargeTransactionDividerGroupType attribute		Required	-

StartTheClockQueryResponse				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
TrackingID	String 12	-	Optional Allows user to retrieve data without requerying again.	-
Choice Block BEGINS			Either QueryResult or QueryError block is returned	
Query Result Block BEGINS	-	-	-	
Sequence Block # 1 BEGINS			1 to many	
LargeTransactionDividerResult	LargeTransactionDividerResult attribute	-	Optional	Refer this attribute block in Appendix A
Sequence Block #2 BEGINS			1 to many	
ClockStarted	clockStartedType complex type	-	Required	Refer to this complex type in Appendix A
ReturnInfo	basicReturnInfo complex type	-	Optional	Refer to this complex type in Appendix A
Sequence Block # 2 ENDS				
Sequence Block # 1 ENDS				
QueryResults Block ENDS				
Query Error Block BEGINS				
ReturnInfo	basicReturnInfo complex type	-	Required	Refer to this complex type in Appendix A
Query Error Block ENDS				
StartTheClockQueryResponse ENDS				

NonComplianceDataWithPostageOwedReportNotification

The purpose of this message is to provide users with notification that their Full-Service Non Compliance data, which may affect their postage statement and FS discounts, is ready to be picked up.

NonComplianceDataWithPostageOwedReportNotification Data Structure

Field Description

NonComplianceDataWithPostageOwedReportNotification				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
NonCOmplianceDataWithPostageOwedReportNotification BEGINS				
SubmittingParty	participantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
PushMessageID	String 25	-	Optional	-
Choice Block BEGINS			Required Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID be provided	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
eDocSenderCRID	CRIDType simple type	-	Optional This is the eDoc Sender's CRID information to be able for *PostalOne!* to track whom actually sent the eDoc data.	Refer to this simple type in Appendix B
FullServiceAvailabilityDate	Date	yyyy-mm-dd	Required	
Notification Date	Date	yyyy-mm-dd	Required	

NonComplianceDataWithPostageOwedReportNotification				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
NonCOmplianceDataWithPostageOwedReportNotification ENDS				

NonComplianceDataWithPostageOwedReportDelivery

This is an automated generated message that is pushed from USPS servers to the user's server, if they have subscribed to a PUSH service. This message pushes the actual Noncompliance data to user's servers.

NonComplianceDataWithPostageOwedReportDelivery Data Structure

Field Description

NonComplianceDataWithPostageOwedReportDelivery				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
NonComplianceDataWithPostageOwedReportDelivery BEGINS				
SubmittingParty	participantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
DataRecipient	dataRecipient complex type	-	Optional	Refer to this complex type in Appendix A
Push Message ID	String 25	-	Optional	-
NonComplianceDataWithPostageOwedReport	NonComplianceDataWithPostageOwedReportType complex type	-	Required 1 to many allowed	Refer to this complex type in Appendix A
LargeTransactionDividerGroupOptionalType	LargeTransactionDividerGroupOptionalType attribute	-	Required	Refer this attribute block in Appendix A
NonComplianceDataWithPostageOwedReportDeliverty ENDS				

NonComplianceDataWithPostageOwedQueryRequest

This message allows the user to query the *PostalOne!* system to retrieve the noncompliance data that may affect the postage statement and FS Discounts.

NonComplianceDataWithPostageOwedQueryRequest Data Structure

Field Description

NonComplianceDataWithPostageOwedQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
NonComplianceDataWithPostageOwedQueryRequest BEGINS				
SubmittingParty	participantIDType complex type	-	Optional	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Optional	Refer to this complex type in Appendix A
Choice Block BEGINS			Required Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID or CustomerAccount block is required	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				

NonComplianceDataWithPostageOwedQueryRequest				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
eDocSenderCRID	CRIDType simple type	-	Optional This is the eDoc Sender's CRID information to allow *PostalOne!* to track whom actually sent the eDoc data.	Refer to this simple type in Appendix B
Start Date	Date	-	Required	-
EndDate	Date	-	Required	-
NonComplianceDataWithPostageOwedQueryRequest ENDS				

NonCompliancePostageOwedQueryResponse

This is a response message to the NonCompliancePostageOwedQueryRequest message that provides the actual data to the requestor.

NonCompliancePostageOwedQueryResponse Data Structure

Field Description

NonCompliancePostageOwedQueryResponse				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
NonCompliancePostageOwedQueryResponse BEGINS				
LargeTransaction DividerGroupType	LargeTransaction DividerGroup Type Attribute block	-	Required	Refer this attribute block in Appendix A
TrackingID	String 12	-	Optional Allows the user to retrieve the data without re-querying	-

NonCompliancePostageOwedQueryResponse				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
Choice Block BEGINS			Either QueryResult or QueryError block is returned	
QueryResult Block BEGINS			Required	
LargeTransactionDividerResult	LargeTransactionDividerResult attribute	-	Optional	Refer this attribute block in Appendix A
NonComplianceDataWithPostageOwedReport	NonComplianceDataWithPostageOwedReportType		Required 1 to many allowed	Refer to this complex type in Appendix A
QueryResult Block ENDS				
QueryError Block BEGINS				
ReturnInfo	basicReturnInfo complex type	-	Required	Refer to this complex type in Appendix A
QueryError block ENDS				
Choice Block ENDS				
NonCompliancePostageOwedQueryResponse ENDS				

USPSMIDCreateRequest

The purpose of this message is to send a request for creating Mailer ID

Prerequisites

- Mailer requests the creation of one or more MIDs.
- Mailer passes in the following information at a minimum per MID requested (information must be repeated for each MID requested):
  - CRID
  - ApplyingForSelf
- PostalOne! will provide feedback to the mailer using USPSMIDCreateResponse Message
- If successful:
  - Response is returned to mailer
  - Response will include per MID requested:
    a) One 9-digit MID if newly created
    b) One or more MIDs if existing

If not successful:

- Response is returned to mailer:
- Response will indicate the problem per MID requested

Business Rules

For each USPSMIDCreateRequest message, the customer needs to provide the information for all of the required blocks.

Submitting Party
Submitting Software
Customer CRID
Customer Name (Optional_
Optional to provide Permit Info
Company HQ Indicator – The accepted values are ‘Yes’ or ‘No’ (Optional)
Address Information (optional)
Sequence Number (Optional)
Applying for Self Indicator. Accepted values are ‘Yes’ or ‘No’ - Required
Required to provide Legal Acknowledgment Block when Self Indicator is ‘No’. The block requires following info:
UnderstandLegalLiabilityForApplyingForSomeOneElse
AcknowledgementNotifyingTheMailOwner
USPSLegalAgreement

USPSMIDCreateRequest Data Structure

Field Description

USPSMIDCreateRequest				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
USPSMIDCreateRequest BEGINS				
SubmittingParty	participantIDType complex type	-	Required	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Required	Refer to this complex type in Appendix A
MIDEntry **USPSMIDCreateRequest ENDS**	midEntryType complex type	-	Required 1 to many	Refer to this complex type in Appendix A

USPSMIDCreateResponse

The message is sent by USPS to customer in response to MID Create request

Prerequisites

- Mailer requests the creation of one or more MIDs.
- Mailer passes in the following information at a minimum per MID requested (information must be repeated for each MID requested): :
    - CRID
    - ApplyingForSelf
- *PostalOne!* will provide feedback to the mailer using USPSMIDCreateResponse Message

If successful:

- Response is returned to mailer
- Response will include per MID requested:

    One 9-digit MID if newly created
    One or more MIDs if existing

If not successful:

- Response is returned to mailer:
- Response will indicate the problem per MID requested

Business Rules

The USPSMIDCreateResponse message returns one of the two messages block i.e. MID Accepted or MID Reject Block.

In ACCEPT block

Returns all of the data blocks/elements that were sent in the USPSMIDCreateRequest message AND

MID information, which is either MID6 or MID9 AND

OPTIONAL return info block that contains return code and return description

in REJECT block

Returns all of the data blocks/elements that were sent in the USPSMIDCreateRequest message AND

OPTIONAL return info block that contains return code and return description to communicate the issue

USPSMIDCreateResponse Data Structure

Field Description

USPSMIDCreateResponse				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
USPSMIDCreateResponse BEGINS				
TrackingID	String 12	-	Optional Allows the	

USPSMIDCreateResponse				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
			user to retrieve the data without querying again.	
Choice Block BEGINS			1 to many allowed Either Accept or Reject block is returned	
USPSMIDCreateAccept Block BEGINS				
MID	MIDType complex type	-	Required 1 to many	Refer to this complex type in Appendix A
MIDStatus	mIDStatus Type simple type	-	Required	Refer to this simple type in Appendix B
MIDEntry	midEntryType complex type	-	Required	Refer to this complex type in Appendix A
ReturnInfo	basicReturnInfo complex type		Optional	Refer to this complex type in Appendix A
USPSMIDCreateAccept Block ENDS				
USPSMIDCreateReject Block BEGINS				
MIDEntry	midEntryType complex type	-	Required	Refer to this complex type in Appendix A
ReturnInfo	basicReturnInfo complex type	-	Required	Refer to this complex type in Appendix A
USPSMIDCreateReject Block ENDS				
Choice Block ENDS				
USPSMIDCreateResponse ENDS				

USPSCRIDCreateRequest

The purpose of this message is to send a request for creating Mailer ID

Prerequisites

- Mailer requests the creation of one or more CRIDs using USPSCRIDCreateRequest Message.
- Mailer passes the required information per CRID requested
- *PostalOne!* provides feedback to the mailer using USPSCRIDCreateResponse Message

  If successful:
  Response is returned to mailer
  Response will indicate whether the included CRID is new or existing
  There will always be exactly one CRID returned for a given CRID requested

  If not successful:
  Response is returned to mailer:
  Response will indicate the problem per CRID requested

USPSCRIDCreateRequest Data Structure

Field Description

USPSCRIDCreateRequest				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
USPSCRIDCreateRequest BEGINS				
SubmittingParty	participantIDType complex type	-	Required	Refer to this complex type in Appendix A
SubmittingSoftware	submittingSoftwareType complex type	-	Required	Refer to this complex type in Appendix A
CRIDEntry	CRIDEntryType complex type	-	Required 1 to many allowed	Refer to this complex type in Appendix A
USPSCRIDCreateRequest ENDS				

USPSCRIDCreateRespone

This message is sent by USPS to customer with the CRID number

Prerequisites

- Mailer requests the creation of one or more CRIDs using USPSCRIDCreateRequest Message.
- Mailer passes the required information per CRID requested
- *PostalOne!* provides feedback to the mailer using USPSCRIDCreateResponse Message

  If successful:

    Response is returned to mailer
    Response will indicate whether the included CRID is new or existing
    There will always be exactly one CRID returned for a given CRID requested

  If not successful:

    Response is returned to mailer:
    Response will indicate the problem per CRID requested

Business Rules

The USPSCRIDCreateResponse message returns one of the two messages block i.e. MID Accepted or MID Reject Block.

1. In ACCEPT block
   - Returns all of the data blocks/elements that were sent in the USPSCRIDCreateRequest message AND
   - CRID number AND
   - OPTIONAL return info block that contains return code and return description to communicate the issues
2. In REJECT block
   - Returns all of the data blocks/elements that were sent in the USPSCRIDCreateRequest message AND
   - OPTIONAL return info block that contains return code and return description to communicate the issue

USPSCRIDCreateRespone Data Structure

Field Description

USPSCRIDCreateRespone				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
USPSCRIDCreate Response BEGINS				
TrackingID	String 12	-	Optional Allows the user to retrieve the data without re-querying it.	-

USPSCRIDCreateRespone				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
Choice Block BEGINS			Required 1 too many allowed. Either QueryResults or QueryError block is returned	
USPSCRIDCreate Accept BEGINS			Required	
CRID	CRIDType	-	Required	Refer to this simple type in Appendix B
CRIDStatus	cRIDStatusType simple type	-	Required	Refer to this simple type in Appendix B
CRIDEntry	cridEntryType complex type	-	Required	Refer to this complex type in Appendix A
ReturnInfo	basicReturnInfo complex type	-	Optional	Refer to this complex type in Appendix A
USPSCRIDCreate Accept Block ENDS				
USPSCRIDCreate Reject Block BEGINS				
CRIDEntry	cridEntryType complex type		Required	Refer to this complex type in Appendix A
ReturnInfo	basicReturnInfo complex type		Required	Refer to this complex type in Appendix A
USPSCRIDCreate Reject Block ENDS				
USPSCRIDCreate Response ENDS				

Appendix A – Mail.XML 11.0D Complex and Attribute Groups Definitions

Complex Type: AddressType

Complex Type: AddressType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
addressType BEGINS				
CommunicationRole (as attribute)	communicationRoleType simple type	-	Optional Attribute	Refer to this simple type in Appendix B
Address1	String 64		Required	
Address2	String 64		Optional	
Address3	String 40		Optional	
Address4	String 40		Optional	
City	String 45		Required	
County	String 45		Optional	
State	stateCode simple type	-	Required	Refer to this simple type in Appendix B
ZipCode	zipCode simple type		Required	Refer to this simple type in Appendix B
GPSCoordinates	gPSCoordinates complex type	-	Optional	Refer to this complex type in Appendix A
addressType ENDS				

Complex Type: addressCorrectionAddressType

Complex Type: addressCorrectionAddressType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
addressCorrectionAddressType BEGINS				
AddressType	addressType simple type	-	Optional	Refer to this simple type in Appendix B
UrbanizationName	String 28	-	Optional	-
PrimaryNumber	String 10	-	Optional	-
PreDirectional	String 2	-	Optional	-
StreetName	String 28	-	Optional	-
StreetSuffix	String 4	-	Optional	-

Complex Type: addressCorrectionAddressType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
PostDirectional	String 2	-	Optional	-
UnitDesignator	String 4	-	Optional	-
SecondaryNumber	String 10	-	Optional	-
City	String 28	-	Optional	-
State	stateCode simple type	-	Optional	Refer to this simple type in Appendix B
Choice block BEGINS			Optional Choice block. Either ZipCode and ZipCodePlus4 OR DeliveryPoint11 DigitZip is required.	
ZipCode	Numeric String 5	-	Required	This element will be return when *PostalOne!* Received 5 digit ZipCode from SASP
ZipCodePlus4	Numeric String 4	-	Optional	When Nine digit ZipCode is received from SASP
DeliveryPoint11DigitZip	Numeric String 11	-	Required	When 11 Digit ZipCode is received from SASP
Choice Block ENDS				
addressCorrectionAddressType ENDS				

Complex: ByForConflictType

Complex: ByForConflictType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**

Choice Block BEGINS			Optional Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID be provided	
UserLicenseCode	userLicenseCodeType simple type	-	Required	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required	Refer to this simple type in Appendix B
CustomerGroupID	String 25	-	Optional	-
MailingGroupID	nonNegativeInteger	-	Required	-
Choice Block ENDS				
MailOwnerConflict Block BEGINS			Optional. 0 to many allowed	
Choice Block BEGINS			One of the following blocks is required. ▪ MailOwner CRID ▪ MailOwner MailerID6 ▪ MailOwnerID9 ▪ PermitInfo block	
MailOwnerCRID	CRIDType simple type	-	Required	Refer to this simple type in Appendix B
MailOwnerMailerID6	mailerID6Type simple type	-	Required	Refer to this simple type in Appendix B
MailOwnerMailerID9	mailerID9Type simple type	-	Required	Refer to this simple type in Appendix B
PermitInfo	permitPublicationDataType complex type	-	Required	Refer to this complex type in Appendix A
Choice Block ENDS				

MailOwnerConflictError block BEGINS			Required 1 to many allowed	
Choice Block BEGINS			Required Maximum 2 occurrence allowed	
MailOwnerCRID	CRIDType simple type	-	Required	Refer to this simple type in Appendix B
MailOwnerMailerID6	mailerID6Type simple type	-	Required	Refer to this simple type in Appendix B
MailOwnerMailerID9	mailerID9Type simple type	-	Required	Refer to this simple type in Appendix B
PermitInfo	permitPublicationDataType complex type	-	Required	Refer to this complex type in Appendix A
Choice Block ENDS				
ConflictReasonCode	String 2	-	Required	-
ConflictReasonDescription	xs:string	-	Required	-
MailOwnerConflictError block ENDS				
MailOwnerConflict block ENDS				
MailPreparerConflict Block BEGINS			Optional. 0 to many allowed	
Choice Block BEGINS			One of the following blocks is required. ▪ MailOwner CRID ▪ MailOwner MailerID6 ▪ MailOwnerID9 ▪ PermitInfo block	
MailPreparerCRID	CRIDType simple type	-	Required	Refer to this simple type in Appendix B
MailPreparerMailerID6	mailerID6Type simple type	-	Required	Refer to this simple type in Appendix B
MailPreparerMailerID9	mailerID9Type simple type	-	Required	Refer to this simple type in Appendix B

PermitInfo	permitPublicationDataType complex type	-	Required	Refer to this complex type in Appendix A
Choice Block ENDS				
MailPreparerConflictError block BEGINS			Required 1 to many allowed	
Choice Block BEGINS			Required Maximum 2 occurrence allowed	
MailPreparerCRID	CRIDType simple type	-	Required	Refer to this simple type in Appendix B
MailPreparerMailerID6	mailerID6Type simple type	-	Required	Refer to this simple type in Appendix B
MailPreparerMailerID9	mailerID9Type simple type	-	Required	Refer to this simple type in Appendix B
PermitInfo	permitPublicationDataType complex type	-	Required	Refer to this complex type in Appendix A
Choice Block ENDS				
ConflictReasonCode	String 2	-	Required	-
ConflictReasonDescription	xs:string	-	Required	-
MailPreparerConflictError block ENDS				
MailPreparerConflict block ENDS				
FullServiceAvailabilityDate	Date	-	Optional	-
ReturnInfo	basicReturnInfo complex type	-	Optional	Refer to this complex type in Appendix A

Attribute: CastOfCharacterResponseHeaderInfo

Attribute: CastOfCharacterResponseHeaderInfo				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Original Sender Msg ID	String, 12	-	Required	-
Receiver Msg ID	String, 12	-	Required	-

Complex Type: ContainerVisibilityEntry

Complex Type: ContainerVisibilityEntry				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
ContainerVisibilityEntry BEGINS				

Complex Type: ContainerVisibilityEntry				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Choice Block BEGINS	-	-	Optional	-
UserLicenseCode	userLicenseCodeType simple type	-	Required when Mail.dat is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when Mail.dat is used	Refer to this simple type in Appendix B
CustomerGroupID	String 25	-	Optional	-
MailingGroupID	nonNegativeInteger	-	Required	-
Choice Block ENDS				
ConsigneeApptID	String 12	-	Optional	-
LogicalIndicator	mailxml_base:yesNo		Optional	
CSAID	mailxml_base:s10		Optional	
ContainerID	String 6		Optional	Refer to this simple type in Appendix B
ParentContainerID	String 6		Optional	Refer to this simple type in Appendix B
SiblingContainerID	String 6		Optional	Refer to this simple type in Appendix B
ContainerType	mailxml_base:mailXMLContainerType		Optional	Refer to this simple type in Appendix B
Sequence Block BEGINS				
USPSInductionFacilityLocaleKey	localeKeyType simple type	-	Optional	Refer to this simple type in Appendix B
USPSInductionFacilityName	String 50	-	Optional	-
ContainerScanDateTime	xs:dateTime	yyyy-mm-ddTHH:MM:SS	Required	-
ScanStatus	inductionStatusType simple type	-	Optional	Refer to this simple type in Appendix B
ScanIssueDescription	mailxml_base:s50	-	Optional	-
ScanIssueResolution	mailxml_base:s50	-	Optional	-
Sequence Block ENDS				
Choice Block BEGINS				
IMcb	mailxml_base:IMcbType		Optional	
IMtb	mailxml_base:IMtbType		Optional	
IMpb	mailxml_base:IMpbType		Optional	
Choice Block ENDS				

Complex Type: ContainerVisibilityEntry				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
FullServiceCompliance Indicator	fullServiceComplianceIndicatorType simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceNon ComplianceReasonCode	reasonCodeType simple type	-	Optional	Refer to this simple type in Appendix B
FSNonFSAvailabilityDate	Date	-	Optional	-
FullServiceIndicator	mailxml_base:yesNo		Optional	
ContainerVisibilityEntry ENDS				

Complex Type: clockStartedType

Complex Type: clockStartedType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
clockStartedType BEGINS				
StarttheclockBEMUBlock BEGINS	-	-	Optional 0 to many allowed	-
Choice Block BEGINS			Optional Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID be provided	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-

MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
PublicationNumber	String 8	-	Optional	-
MailClass	mailClassType simple type	-	Optional	Refer to this simple type in Appendix B
ProcessingCategory	processingCategoryType simple type	-	Optional	Refer to this simple type in Appendix B
OriginFacilityName	String 50	-	Optional	
OriginFacilityLocaleKey	localeKeyType simple type	-	Optional	Refer to this simple type in Appendix B
ArrivalDateTime	xs:dateTime	yyyy-mm-ddTHH:MM:SS	Optional	-
StartTheClockDate	xs:date	yyyy-mm-dd	Required	-
FullServiceCompliance Indicator	fullServiceComplianceIndicatorType simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceNonCompliance ReasonCode	reasonCodeType simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceAvailabilityDate	Date	yyyy-mm-dd	Optional	-
Sequence Block BEGINS			Optional	
ContainerID	String 6	-	Optional 0 to many allowed	-
ContainerType	mailXMLContainerType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B

Choice Block BEGINS			Optional Only one of the following can be provided IMcb IMtb IMpb	
IMcb	IMcbType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B
IMtb	IMtbType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B
IMpb	IMpbType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B
Choice Block ENDS				
Sequence Block ENDS				
StarttheclockBEMUBlock End				
StarttheclockDropShipOrOrigin Block BEGINS			Optional 0 to many allowed	
Choice Block BEGINS			Optional Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID be provided	

UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
Choice Block BEGINS			Optional Either ConsigneeApptID or CSADispatchNumber is returned	
ConsigneeApptID	Mailxml_base:s10	String 10	Required when CSA Dispatch Number not provided	-
CSADispatchNumber	String 10	-	Required when ConsigneeApptID is not provided	-
Choice Block ENDS				
PublicationNumber	String 8	-	Optional	-
MailClass	mailClassType simple type	-	Optional	Refer to this simple type in Appendix B
ProcessingCategory	processingCategoryType simple type	-	Optional	Refer to this simple type in Appendix B

DropShipYesNo	yesNo simple type	-	Required	Refer to this simple type in Appendix B
UspsInductionFacilityName	String 50	-	Optional	-
UspsInductionFacilityLocaleKey	localeKeyType simple type	-	Optional	Refer to this simple type in Appendix B
ApptDateTime	xs:dateTime	yyyy-mm-ddTHH:MM:SS	Optional	-
ArrivalDateTime	xs:dateTime	yyyy-mm-ddTHH:MM:SS	Optional	-
UnloadStartDateTime	xs:dateTime	yyyy-mm-ddTHH:MM:SS	Optional	-
UnloadEndDateTime	xs:dateTime	yyyy-mm-ddTHH:MM:SS	Optional	-
ApptStatus	apptStatusType simple type	-	Optional	Refer to this simple type in Appendix B
StartTheClockDate	xs:date	yyyy-mm-dd	Optional	-
FullServiceComplianceIndicator	fullServiceComplianceIndicatorType simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceNonComplianceReasonCode	reasonCodeType simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceAvailabilityDate	xs:date	yyyy-mm-dd	Optional	-
Sequence Block BEGINS			Optional	
ContainerID	String 6	-	Optional 0 to many allowed	-
ContainerType	mailXMLContainerType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B

Choice Block BEGINS			Optional Only one of the following can be provided IMcb IMtb IMpb	
IMcb	IMcbType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B
IMtb	IMtbType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B
IMpb	IMpbType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B
Choice Block ENDS				
Sequence Block ENDS				
StarttheclockDropShipOrigin Block ENDS				
StarttheclockPlantLoad Block BEGINS			Optional 0 to many allowed	
Choice Block BEGINS			Optional Either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID be provided	

UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
CSADispatchNumber	String 10	-	Optional	-
PublicationNumber	String 8	-	Optional	-
MailClass	mailClassType simple type	-	Optional	Refer to this simple type in Appendix B
ProcessingCategory	processingCategoryType simple type	-	Optional	Refer to this simple type in Appendix B
OriginFacilityName	String 50	-	Optional	
OriginFacilityLocaleKey	localeKeyType simple type	-	Optional	Refer to this simple type in Appendix B
MailReleasedDateTime	DateTime	yyyy-mm-ddTHH:MM:SS	Optional	-
StartTheClockDate	Date	yyyy-mm-dd	Optional	-
FullServiceCompliance Indicator	fullServiceComplianceIndicatorType simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceNonCompliance ReasonCode	reasonCodeType simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceAvailabilityDate	Date	yyyy-mm-dd	Optional	-
Sequence Block BEGINS			Optional	

ContainerID	String 6	-	Optional 0 to many allowed	-
ContainerType	mailXMLContainerType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B
Choice Block BEGINS			Optional Only one of the following can be provided IMcb IMtb IMpb	
IMcb	IMcbType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B
IMtb	IMtbType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B
IMpb	IMpbType simple type	-	Optional 0 to many allowed	Refer to this simple type in Appendix B
Choice Block ENDS				
Sequence Block ENDS				
StarttheclockPlantLoad Block ENDS				
clockStartedType ENDS				

Complex Type: CofCCreateRejectType

Complex Type: CofCCreateRejectType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**

Complex Type: CofCCreateRejectType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Choice Block BEGINS			Choice is to either to return the MailPiece Group or Container Group block data or job info	
Choice Block BEGINS				
User License Code	String		Required when 11.0D version is used	-
Maildat Job ID	String	-	Required when 11.0D version is used	-
Customer Group ID	String, 25 character	-	Required when Mail.XML version is used	-
Mailing Group ID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
MailPieceGroup	Complex	-	Required System can return 1 to many MailPiece Group block data	Refer ‘mailPieceGroup Create Type’ complex in appendix A
ContainerGroup			Required system can return 1 to many Container Group block data	Refer ‘group Container IDType’ complex in appendix A
Choice Block ENDS				
ReturnInfo	Complex	-	Optional	Refer ‘ReturnInfo’ complex in appendix A
Sequence Block ENDS				

Complex Type: containerGroupType

Complex Type: containerGroupType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Attribute Block BEGINS				
SenderMailPieceGroupID	String 12	-	-	-
ReceiverMailPieceGroupID	String 12	-	-	-

Complex Type: containerGroupType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Attribute Block ENDS				

Complex Type: ContainerErrorWarningBlockType

Complex Type: ContainerErrorWarningBlockType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
ContainerErrorWarningBlockType BEGINS				
ConsigneeContainerGroupingID	String 16		Optional	-
ConsigneeContentID	String 12	-	Optional	-
ShipperContainerGroupingID	String 16	-	Optional	-
IMcb	IMcbType simple type	-	Optional	Refer to this simple type in Appendix B
IMtb	IMtbType simple type	-	Optional	Refer to this simple type in Appendix B
IMpb	IMpbType simple type	-	Optional	Refer to this simple type in Appendix B
MailContentName	String 30	-	Optional	-
SchedulerContentID	String 12	-	Optional	-
ContainerErrorWarningBlockType ENDS				

Complex Type: CRIDEntryType

Complex Type: CRIDEntryType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
CRIDEntryType BEGINS				
CompanyName	String 40		Required	-
PermitPublicationData	permitPublicationDataType complex type	-	Optional	Refer to this complex type in Appendix A
CompanyHQIndicator	yesNo simple type	-	Optional	Refer to this simple type in Appendix B

Complex Type: CRIDEntryType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
Address	addressType complex type	-	Required	Refer to this complex type in Appendix A
AddressMatch	yesNo simple type	-	Required	Refer to this simple type in Appendix B
ApplyingForSelf	yesNo simple type	-	Required	Refer to this simple type in Appendix B
LegalAcknowledgementBlock	legalAcknowledgementBlockCRIDtype complex type	-	Optional	Refer to this complex type in Appendix A
CRIDEntryType ENDS				

Complex Type: DataRecipient

Complex Type: DataRecipient				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
DataRecipient Block BEGINS				
Sequence Block BEGINS				
CRID	CRIDType simple type	-	Required	Refer to this simple type in Appendix B
Roles	roleType simple type	-	Required	Refer to this simple type in Appendix B
Sequence Block ENDS				
DataRecipient Block ENDS				

Complex Type: ForeignAddressType

Complex Type: ForeignAddressType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
ForeignAddressType Block BEGINS				
Sequence Block BEGINS				
AddressLine1	String 37		Optional	
AddressLine2	String 37		Optional	

Complex Type: ForeignAddressType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
AddressLine3	String 37		Optional	
CountryCode	String 3	-	Optional	-
Sequence Block ENDS				
ForeignAddressType Block ENDS				

Complex Type: NonComplianceDataWithPostageOwedReportType

Complex Type: NonComplianceDataWithPostageOwedReportType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
NonComplianceDataWithPostageOwedReportType BEGINS				
Choice Block BEGINS			Required block. Either ULC/MaildatJobID OR CustomerGroupID/MailingGroupID are required.	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				
eDocSenderCRID	CRIDType simple type	-	Optional	Refer to this simple type in Appendix B
Sequence Block BEGINS			Optional 0 to many allowed	

Complex Type: NonComplianceDataWithPostageOwedReportType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
TotalFSPieces	nonNegativeInteger	-	Optional	-
TotalAffectedPieces	nonNegativeInteger	-	Optional	-
TotalDiscountAssessment	Double	-	Optional	-
MailClass	mailClassType simple type	-	Optional	Refer to this simple type in Appendix B
DiscountFinalized	yesNo simple type	-	Optional Yes indicates that discount has been finalized.	Refer to this simple type in Appendix B
Sequence Block ENDS				
MailingIDDetail Block BEGINS			Optional 0 to many allowed	
ContainerID	String 6	-	Optional	-
ContainerType	mailXMLContainerType simple type	-	Optional	Refer to this simple type in Appendix B
SiblingRefContainerID	String 6	-	Optional	-
ParentContainerID	String 6	-	Optional	-
IMcb	IMcbType simple type	-	Optional	Refer to this simple type in Appendix B
IMtb	IMtbType simple type	-	Optional	Refer to this simple type in Appendix B
Sequence Block BEGINS			Optional 0 to many allowed	
VerificationReportType	mailxml_dd:verificationErrorType	-	Optional	Refer to this simple type in Appendix B
TotalAffectedPieceCnt	nonNegativeInteger	-	Optional	-
VerificationReportType	mailxml_dd:verificationErrorType	-	Optional	Refer to this simple type in Appendix B
Sequence Block ENDS				
NonComplianceDataWithPostage				

Complex Type: NonComplianceDataWithPostageOwedReportType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
OwedReportType ENDS				

Complex Type: dqrContainerInfoType

Complex Type: dqrContainerInfoType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
dqrContainerInfoType BEGINS				
Sequence Block BEGINS				
ContainerID	String 6		Optional	
SiblingRefContainerID	String 6	-	Optional	-
ParentContainerID	String 6		Optional	
ProcessingCategory	processingCategoryType simple type	-	Optional	Refer to this simple type in Appendix B
MailClass	mailClassType simple type	-	Optional	Refer to this simple type in Appendix B
USPSInductionFacilityLocale Key	localeKeyType simple type	-	Optional	Refer to this simple type in Appendix B
MailingDate	Date	yyyy-mm-dd	Optional	
ContainerType	mailXMLContainerType simple type	-	Optional	Refer to this simple type in Appendix B
IMcb	IMcbType simple type	-	Optional	Refer to this simple type in Appendix B
IMtb	String 25		Optional	
CSAID	String 10		Optional	
ConsigneeApptID	String 12		Optional	
FullServiceAvailabilityDate	Date	-	Optional	-
VerificationError	mailxml_dd:verificationErrorType		Optional 0 to many	Refer to this complex type in Appendix A
Sequence BEGINS			Optional 0 to many allowed	
PreparerCRID	CRIDType simple type		Optional	Refer to this simple type in Appendix B

Complex Type: dqrContainerInfoType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
OwnerCRID	CRIDType simple type		Optional	Refer to this simple type in Appendix B
Sequence ENDS				
Choice Block BEGINS		0 to many Either ieceInfo block or PieceRangeInfo block		
PieceInfo Block BEGINS		0 to many		
PhysicalPieceID	String 22		Required	
MailerPostageAccountID	String 10	-	Optional	-
MPADescription	String 30	-	Optional	-
IMB	iMBType simple type	-	Required	Refer to this simple type in Appendix B
OwnerCRID	CRIDType simple type	-	Required	Refer to this simple type in Appendix B
PreparerCRID	CRIDType simple type	-	Required	Refer to this simple type in Appendix B
VerificationError	mailxml_dd:verificationErrorType	1 to many	Required	Refer to this complex type in Appendix A
PieceInfo Block ENDS				
PieceRangeInfo Block BEGINS			Optional	
PieceRangeID	String 22		Required	
IMBUpperSerialization	iMBUpperSerializationType simple type	-	Required	Refer to this simple type in Appendix B
IMBLowerSerialization	iMBLowerSerializationType simple type	-	Required	Refer to this simple type in Appendix B
MailClass	mailClassType simple type	-	Required	Refer to this simple type in Appendix B
ProcessingCategory	processingCategoryType simple type	-	Optional	Refer to this simple type in Appendix B

Complex Type: dqrContainerInfoType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
PieceCountInRange	nonNegativeInteger	-	Optional	-
AffectedPieceCount	nonNegativeInteger	-	Optional	-
OwnerCRID	CRIDType simple type		Required	Refer to this simple type in Appendix B
PreparerCRID	CRIDType simple type		Required	Refer to this simple type in Appendix B
VerificationError	mailxml_dd:verificationErrorType	1 to many	Required	Refer to this simple type in Appendix B
PieceRangeInfo Block ENDS				
Choice Block ENDS				
Sequence Block ENDS				
dqrContainerInfoType ENDS				

Complex Type: AddressCorrectionType

Complex Type: AddressCorrectionType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
AddressCorrection BEGINS				
Sequence Block BEGINS				
Choice Block BEGINS			Optional	
UserLicenseCode	userLicenseCodeType simple type		Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Required when Mail.XML 11.0D is used	-
Choice Block ENDS				

Complex Type: AddressCorrectionType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Choice Block BEGINS			Optional. When provided; one of the following is required: MailerID6 or MailerID9	
OriginalMailerID6	mailerID6Type simple type	-	Required	Refer to this simple type in Appendix B
OriginalMailerID9	mailerID9Type simple type	-	Required	Refer to this simple type in Appendix B
Choice Block ENDS				
OriginalIMB	iMBType simple type	-	Optional	Refer to this simple type in Appendix B
RecordCreationDate	Date	yyyy-mm-dd	Optional	-
MoveEffectiveDate	Date	yyyy-mm-dd	Optional	-
MoveType	addressCorrectionMoveType simple type	-	Optional	Refer to this simple type in Appendix B
DeliverabilityCodeType	deliverabilityCodeType simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceComplianceIndicator	fullServiceComplianceIndictorType simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceNon ComplianceReasonCode	reasonCodeType simple type	-	Optional	Refer to this simple type in Appendix B
COAName block BEGINS			Optional Block	
Choice Block BEGINS			Either provide Business Name or everything else in this block	
BusinessName	String 50	-	Required	-
LastName	String, Length 50	-	Required	-

Complex Type: AddressCorrectionType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
FirstNameMiddleName	String, Length 15	-	Required	-
Prefix	String, Length 15	-	Optional	
Suffix	String, Length 15	-	Optional	-
Choice Block ENDS				
COAName Block ENDs				
KeylineFromEDoc	String	-	Optional	-
OldAddress	addressCorrectionAddressType complex type		Optional	Refer to this complex type in Appendix A
NewAddress Block BEGINS	-		Optional	
Choice Block BEGINS				
NewDomesticAddress	addressCorrectionAddressType complex type	-	Required when block info is provided and no foreign address	Refer to this complex type in Appendix A
NewForeignAddress	foreignAddressType complex type	-	Required when block info is provided and no domestic address correction	Refer to this complex type in Appendix A
Choice Block ENDS				
NewAddress Block ENDS				
PrivateMailbox	String, Length 37		Optional	-
ClassNotificationType	classNotificationType	-	Optional	Refer to this simple type in Appendix B
FeeNotification	FeeNotificationType simple type	-	Optional	Refer to this simple type in Appendix B
BillReasonCode	billReasonCodeType simple type	-	Optional	Refer to this simple type in Appendix B
FreeChargeFlag	yesNo simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceAvailabilityDate	Date	-	Optional	-

Complex Type: AddressCorrectionType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
ReturnInfo	basicReturnInfo complex type	-	Optional	Refer to this complex type in Appendix A
Sequence Block ENDS				
AddressCorrection ENDS				

Complex Type: NixieDetailType

Complex Type: NixieDetailType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
FullServiceNixieDetailType BEGINS				
Sequence Block BEGINS				
Choice Block BEGINS			Optional Choice block. If provided then: either UserLicenseCode/MaildatJobID or CustomerGroupID/MailingGroupID are required	
UserLicenseCode	userLicenseCodeType simple type		Conditionally Required when 11.0D version is used	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type	-	Conditionally Required when 11.0D version is used	Refer to this simple type in Appendix B
CustomerGroupID	String, 25 character	-	Optional	-
MailingGroupID	Non Negative Integer	-	Conditionally Required when Mail.XML 11.0D is used	-
Choice Block ENDS				

Complex Type: NixieDetailType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Choice Block BEGINS			Optional block. If provided then one of the following element is required: OriginalMailerID6 or OriginalMailerID9	
OriginalMailerID6	mailerID6Type simple type	-	Required.	Refer to this simple type in Appendix B
OriginalMailerID9	mailerID9Type	-	Required	Refer to this simple type in Appendix B
Choice Block ENDS				
OriginalIMB	iMBType simple type	-	Optional	Refer to this simple type in Appendix B
RecordCreationDate	Date	yyyy-mm-dd	Optional	-
ActionCode	actionCodeType simple type	-	Optional	Refer to this simple type in Appendix B
ParsedAddressOnPiece	String	-	Optional	-
OnPieceCityStateZip	String	-	Optional	-
ReturnedToAddress	String	-	Optional	
ReturnedtoCityStateZip	String	-	Optional	-
ReasonCode	reasonCodeType simple type	-	Optional	Refer to this simple type in Appendix B
KeylineFromEDoc	String	-	Optional	-
ClassNotificationType	classNotificationType simple type	-	Optional	Refer to this simple type in Appendix B
FeeNotification	FeeNotificationType simple type	-	Optional	Refer to this simple type in Appendix B

Complex Type: NixieDetailType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
FullServiceComplianceIndicator	fullServiceComplianceIndicatorType simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceNon ComplianceReasonCode	reasonCodeType simple type	-	Optional	Refer to this simple type in Appendix B
BillReasonCode	billReasonCodeType simple type	-	Optional	Refer to this simple type in Appendix B
FreeChargeFlag	yesNo simple type	-	Optional	Refer to this simple type in Appendix B
FullServiceAvailabilityDate	Date	-	Optional	-
ReturnInfo	basicReturnInfo complex type	-	Optional	Refer to this complex type in Appendix A
Sequence Block ENDS				
FullServiceNixieDetailType ENDS				

Complex Type: GPSCoordinates

Complex Type: GPSCoordinates				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
gPSCoordinates BEGINS				
GPSSystem (as attribute)	gPSSystemType simple type	-	Optional Attribute	Refer to this simple type in Appendix B
Latitude	mailxml_base:dec18-8	-	Required	Refer to this simple type in Appendix B
Longitude	mailxml_base:dec18-8	-	Required	Refer to this simple type in Appendix B
Height bock BEGINS	-		Optional	
Value block BEGINS	-	-	Required	

Complex Type: GPSCoordinates				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
UOM	mailxml_Base:UOMType	-	Required Attribute	Refer to this simple type in Appendix B
Value Block ENDS				
RangeMin Block BEGINS	-	-	Optional	
UOM	mailxml_Base:UOMType	-	Required Attribute	Refer to this simple type in Appendix B
RangeMin Block ENDS				
RangeMax Block BEGINS	-	-	Optional	
UOM	mailxml_Base:UOMType	-	Required Attribute	Refer to this simple type in Appendix B
RangeMax Block ENDS				
Height bock ENDS				
gPSCoordinates ENDS				

Complex Type: groupContainerIDType

Complex Type: groupContainerIDType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Attribute Block BEGINS				
SenderMailPieceGroupID	String 12	-	Required	-
ReceiverMailPieceGroupID	String 12	-	Optional	-
Attribute Block ENDS				
Choice Block BEGINS			1 to many	
MailDatContainer	MailDatContainer complex type		Required	Refer to this complex type in Appendix A
Mail.XML Container	mailXMLContainerIDType complex type		Required	Refer to this complex type in Appendix A
ContainerBarcode	IMcbType simple type		Required	Refer to this simple type in Appendix B
Choice Block ENDS				

Complex Type: intelligentMailBarcodeType

Complex Type: intelligentMailBarcodeType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
BarcodeID	iMBarcodeIDType	Numeric string 2	Optional	Refer to this simple type in Appendix B
Choice Block BEGINS			Optional	
ServiceTypeCode	ServiceTypeCodeType simple type	Numeric string 3	Required	Refer to this simple type in Appendix B
OR				
MailClass	mailClassType simple type	Refer 'mailClassType' simple type	Required	Refer to this simple type in Appendix B
Choice Block ENDS				
Choice Block BEGINS				
MailerID6	mailerID6Type	String 6	Required	Refer to this simple type in Appendix B
SerialNumber9	iMBSerialNumber9Type	Numeric string 9	Required	Refer to this simple type in Appendix B
OR				
MailerID9	mailerID9Type	String 9	Required	Refer to this simple type in Appendix B
SerialNumber6	:iMBSerialNumber6Type	Numeric string 6	Required	Refer to this simple type in Appendix B
Choice Block ENDS				
DeliveryPointZIP	RoutingZipCodeType	String 11	Optional	Refer to this simple type in Appendix B

Attribute Group: LargeTransactionDividerResult

Attribute Group: LargeTransactionDividerResult				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
LargeTransactionDividerResult BEGINS	-	-	-	-
TrackingID	String 12	-	Required	-
LargeTransactionDividerResult ENDS	-	-	-	-

Attribute Group: LargeTransactionDividerGroupType

Attribute Group: LargeTransactionDividerGroupType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**

Attribute Group: LargeTransactionDividerGroupType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
LargeTransactionDividerGroupType BEGINS				
MessageGroupID	String, Length 20	-	Required	Refer to this simple type in Appendix B
TotalMessageCount	Positive Integer	-	Required	Refer to this simple type in Appendix B
MessageSerialNumber	Positive Integer	-	Required	Refer to this simple type in Appendix B
TransmittedRecordCount	Positive Integer	-	Required	Refer to this simple type in Appendix B
TotalRecordsAcrossMessages	Positive Integer	-	Required	Refer to this simple type in Appendix B
LastMessage	yesNo simple type	-	Optional Default is set to "NO"	Refer to this simple type in Appendix B
LargeTransactionDividerGroupType ENDS				

Attribute Group: LargeTransactionDividerGroupOptionalType

Attribute Group: LargeTransactionDividerGroupOptionalType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
LargeTransactionDividerGroupOptionalType BEGINS				
Sequence Block BEGINS				
MessageGroupID	String, Length 20	-	Optional	-
TotalMessageCount	Positive Integer	-	Optional	-
MessageSerialNumber	Positive Integer	-	Optional	-
TransmittedRecordCount	Positive Integer	-	Optional	-
TotalRecordsAcrossMessage	Positive Integer	-	Optional	-

Attribute Group: LargeTransactionDividerGroupOptionalType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
LastMessage	yesNo simple type	-	Optional Default is set to "NO"	Refer to this simple type in Appendix B
Sequence Block ENDS				
LargeTransactionDividerGroupOptionalType ENDS				

Complex Type: LegalAknowledgmentBlockCRIDType

Complex Type: LegalAknowledgmentBlockCRIDType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
LegalAcknowledgementBlockCRIDType BEGINS				
UnderstandLegalLiabilityForApplyingForSomeOneElse	yesNo simple type	-	Required	Refer to this simple type in Appendix B
AcknowledgeNotifyingTheMailOwner	yesNo simple type	-	Required	Refer to this simple type in Appendix B
USPSLegalAgreement	String		**Required.** * There is some default language	-
LegalAcknowledgementBlockCRIDType ENDS				

Complex Type: LegalAknowledgmentBlockMIDType

Complex Type: LegalAknowledgmentBlockMIDType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
LegalAcknowledgementBlockMIDType BEGINS				
UnderstandLegalLiabilityForApplyingForSomeOneElse	yesNo simple type	-	Required	Refer to this simple type in Appendix B
AcknowledgeNotifyingTheMailOwner	yesNo simple type	-	Required	Refer to this simple type in Appendix B

Complex Type: LegalAknowledgmentBlockMIDType				
Field	Format	Acceptable Value	Business Rules	Comments
USPSLegalAgreement	String		**Required.** * There is some default language	-
LegalAcknowledgementBlockMIDType ENDS				

Complex Type: mailDatContainerIDType

Complex Type: mailDatContainerIDType				
Field	Format	Acceptable Values	Business Rules	Comments
mailDatContainerIDType BEGINS				
Sequence Block BEGINS				
UserLicenseCode	userLicenseCodeType simple type		required	Refer to this simple type in Appendix B
MaildatJobID	jobIDType simple type		required	Refer to this simple type in Appendix B
ContainerID	String 6		required	
UniqueContainerBarcode	IMcbType simple type		Optional	Refer to this simple type in Appendix B
Sequence Block ENDS				
mailDatContainerIDType ENDS				

Complex Type: mailPieceGroupType

Complex Type: mailPieceGroupType				
Field	Format	Acceptable Values	Business Rules	Comments
Attribute Block BEGINS				
SenderMailPieceGroupID	String 12	-	-	-
ReceiverMailPieceGroupID	String 12	-	-	-
Attribute Block ENDS				

Complex Type: mailPieceGroupCreateType

Complex Type: mailPieceGroupCreateType				
Field	Format	Acceptable Values	Business Rules	Comments
Attribute Block BEGINS				

Complex Type: mailPieceGroupCreateType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
SenderMailPieceGroupID	String 12	-	-	-
ReceiverMailPieceGroupID	String 12	-	-	-
Attribute Block ENDS				
Choice Block BEGINS			1 to many	
MailPieceID	mailPieceIDType complex type		Required	Refer 'mailPiece IDType' complex in Appendix A
Sequence Block BEGINS				
LowestMailPieceID	mailPieceIDType complex type		Required	Refer 'mailPiece IDType' complex in Appendix A
HighestMailPieceID	mailPieceIDType complex type	-	Required	Refer 'mailPiece IDType' complex in Appendix A

Complex Type: mailPieceIDType

Complex Type: mailPieceIDType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
Choice Block BEGINS			Optional	
IMB	intelligentMailBarcodeType complex type		Required	Refer 'intelligent MailBarcode Type' complex in Appendix A
OR				
PlanetCode	planetCodeType simple type	Min 11 and Max 13 characters string	Optional	Refer to this simple type in Appendix B
DeliveryPointZIP	RoutingZipCodeType	String 11	Required	Refer to this simple type in Appendix B
OR				
DeliveryContainerID	mailxml_base:ns22	Numeric String 22	Required	
DeliveryPointZIP	RoutingZipCodeType	String 11	Required	Refer to this simple type in Appendix B
choice Block ENDS				

Complex Type: mailXMLContainerIDType

Complex Type: mailXMLContainerIDType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
mailXMLContainerIDType BEGINS				
Choice Block BEGINS				
MailingGroupID	Non Negative Integer		required	
CustomerGroupID	String 25		Required	
ContainerID	String 6		Required	
ContainerBarcode	IMcbType simple type	-	Optional	Refer to this simple type in Appendix B
Choice Block ENDS				
mailXMLContainerIDType ENDS				

Attribute: mailXMLHeaderInfo

Attribute: mailXMLHeaderInfo				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
mailXMLHeaderInfo BEGINs				
Sequence Block BEGINS				
User License Code	String		Optional	-
Maildat Job ID	String	-	Optional	-
Customer Group ID	String, 25 character	-	Optional	-
Mailing Group ID	Non Negative Integer	-	Optional	-
Sequence Block ENDS				
mailXMLHeaderInfo ENDS				

Complex: MidEntryType

Complex: MidEntryType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
midEntryType BEGINS				
Sequence Block BEGINS				
CustomerCRID	CRIDType simple type	-	Required	Refer to this simple type in Appendix B

Complex: MidEntryType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
CustomerName	String		Optional	
PermitPublicationData	permitPublicationDataType complex type	-	Optional	Refer to this complex type in Appendix A
CompanyHQIndicator	yesNo simple type	-	Optional	Refer to this simple type in Appendix B
Address1	String 64		Optional	
Address2	String 64		Optional	
City	String 50		Optional	
State	String 2		Optional	
ZipCode	Numeric String 5		Optional	
SequenceNumber	Integer		Optional	
ApplyingForSelf	yesNo simple type	-	Required	Refer to this simple type in Appendix B
LegalAcknowledgment Block	legalAcknowledgementBlockMIDType complex type	-	Optional	Refer to this complex type in Appendix A
Sequence Block ENDS				
midEntryType ENDS				

Complex: MIDType

Complex: MIDType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
MIDType BEGINS				
Sequence Block BEGINS				
MID6	mailerID6Type simple type		Required	Refer to this simple type in Appendix B
OR				
MID9	mailerID9Type simple type		Required	Refer to this simple type in Appendix B
Sequence Block ENDS				
MIDType ENDS				

Complex: MPSRequestTypeType

Complex: MIDType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
MPSRequestTypeType				

Complex: MIDType				
BEGINS				
ScanState	mailxml_dd:MPSStateType		Required	Refer to this simple type in Appendix B
MPSQueryBy	mailxml_dd:MPSQueryType		Required	Refer to this simple type in Appendix B
MPSRequestTypeType ENDS				

Complex Type: permitPublicationDataType

Complex Type: permitPublicationDataType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
permitPublicationDataType BEGINS				
Choice Block BEGINS	-	-	Either PermitNumber, PermitType, PermitZip4 OR Publication Number is required	-
Sequence Block BEGINS				
Permit Number	String, 8	-	Required, when providing Permit Number and Permit Zip4 data	-
Permit Type	permitTypeType simple type	-	Required	Refer to this simple type in Appendix B
Permit Zip4	Numeric String, 9	-	Required when providing Permit Number and Permit Zip 4 data	Refer to this simple type in Appendix B
Sequence Block ENDS				
Publication Number	String, 8	-	Either provide Permit Number, Permit Type 4 and optional Permit Type OR this field.	-
Choice Block ENDS				
permitPublicationDataType ENDS				

Complex Type: basicReturnInfo

Complex Type: basicReturnInfo				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
basicReturnInfo BEGINS				
Sequence Block BEGINS			Optional 0 to many allowed	
Return Code	Numeric String – Length 4	-	Optional	
Return Description	String, Length 260	-	Required	
ContainerErrorWarningBlock	containerErrorWarningBlock Type complex type	-	Optional	Refer to this complex type in Appendix A
Sequence Block ENDS				
basicReturnInfo ENDS				

Complex Type: participantIDType

Complex Type: participantIDType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
participantIDType BEGINS				
Sequence Block BEGINS				
MailerID6	mailID6Type simple type		Not required (attribute) Either MailerID6 or MailerID9 can be provided, not both	Refer to this simple type in Appendix B
MailerID9	mailerID9Type		Not required (attribute) Either MailerID6 or MailerID9 can be provided, not both	Refer to this simple type in Appendix B
CRID	CRIDType		Not required (attribute), Either CRID or MID can be provided for authorization	Refer to this simple type in Appendix B
SchedulerID	String, 12		Optional	
MaildatUserLicense	userLicenseCodeType simple type		Optional	Refer to this simple type in Appendix

Complex Type: participantIDType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
				B
ShippingAgentID	String 12	-	Optional	-
ReceivingAgentID	String 12	-	Optional	-
Sequence Block ENDS				
participantIDType ENDS				

Complex Type: SubmittingSoftwareType

Complex Type: SubmittingSoftwareType				
Field	**Format**	**Acceptable Values**	**Business Rules**	**Comments**
submittingSoftwareType BEGINS				
Sequence Block BEGINS				
SoftwareName	String		Required	
Vendor	String		Required	
Version	String		Required	
ApprovalDate	Date	yyyy-mm-dd	Optional	Changed the type to 'Date' based on errata
ApprovalKey	String		Optional	
Sequence Block ENDS				
submittingSoftwareType ENDS				

Complex Type: VerificationErrorType

Complex Type: VerificationErrorType				
Field	**Format**	**Acceptable Value**	**Business Rules**	**Comments**
VerificationErrorType BEGINS				
Sequence Block BEG*I*NS			Optional 0 to many allowed	
VerificationErrorCode	Numeric String 4		Required	

Complex Type: VerificationErrorType				
VerificationError Description	String 260		Optional	
Sequence Block ENDS				
VerificationErrorType ENDS				

Appendix B – Mail.XML 11.0D Simple Types Data Structure

simpleType: actionCodeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	F = Forwarded
enumeration	W = Wasted
enumeration	R = Returned
enumeration	U = Unable to determine action

simpleType: addressCorrectionMoveType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	F = Family Move
enumeration	I = Individual Move
enumeration	B = Business Move

simpleType: additionalInHomeRangeType

Tag	Mail.XML 11.0D
Base	mailxml_base:n01

simpleType: adjustmentStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	R = Ready to Pay
enumeration	X = Paid
enumeration	C = Cancel
enumeration	P = Preliminary Postage Statement
enumeration	T = Transportation

simpleType: adjustmentType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	1 = Re-order in excess of tolerance
enumeration	2 = “Average Remail” Add-on
enumeration	3 = Spoilage
enumeration	4 = Shortage

simpleType: agencyCostCodeType

Tag	Mail.XML 11.0D

Tag	Mail.XML 11.0D
Base	xs:string
pattern	[0-9]{5}|[0-9]{6}

simpleType: apptMethodType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Online
enumeration	Phone
enumeration	Webservices - Detail
enumeration	Webservices - No Detail

simpleType: apptStateType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Scheduled
enumeration	Arrived
enumeration	ApptValidated
enumeration	WaitingToBeUnloaded
enumeration	Unloading
enumeration	FinishedUnloading
enumeration	ResolvingProblems
enumeration	Rejected
enumeration	Closed

simpleType: apptStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Open
enumeration	Closed
enumeration	No Show
enumeration	Unscheduled
enumeration	Rejected

simpleType: apptTypeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Pallet
enumeration	DropAndPick
enumeration	Speedline
enumeration	Bedload

simpleType: barcodeRateType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Y = Yes
enumeration	N = No

Tag	Mail.XML 11.0D
enumeration	O = Other

simpleType: billReasonCodeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	E
enumeration	F
enumeration	M
enumeration	N

simpleType: bundleDetailLevelType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = Firm
enumeration	B = Carrier Route
enumeration	C = 5-Digit
enumeration	D = Unique 3-Digit
enumeration	E = (Reserve for future)
enumeration	F = 3-Digit
enumeration	G = (Reserve for future)
enumeration	H = ADC
enumeration	I = AADC
enumeration	J = (Reserve for future)
enumeration	K = Origin MxADC
enumeration	L = MxADC
enumeration	M = MxAADC
enumeration	N = (Reserve for future)
enumeration	O = Working
enumeration	P = (Reserve for future)
enumeration	Q = (Reserve for future)
enumeration	R = Parcel
enumeration	S = Multi-Piece Parcel
enumeration	T = 3D Scheme
enumeration	U = 5D Scheme + L007
enumeration	V = NDC (previously as BMC)
enumeration	W = 5-Digit Super Scheme
enumeration	X = FSS Sort Plan

simpleType: bundleRecordStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	O = Original
enumeration	D = Delete
enumeration	I = Insert
enumeration	U = Update

simpleType: bundleStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Blank
enumeration	C = Cancel

simpleType: canadaAdMailOnlyType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	ST = Standard
enumeration	LG = Large

simpleType: chargeableBundleLevelType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	3 = 3-Digit/SCF
enumeration	5 = 5-Digit/Carrier Route
enumeration	A = ADC/AADC
enumeration	C = Carrier Route
enumeration	F = Firm
enumeration	M = Mixed ADC and Mixed AADC

simpleType: chargeableContainerType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Sacks-Trays
enumeration	Pallets

simpleType: chargeableContainerLevelType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	3 = 3-Digit/SCF
enumeration	5 = 5-Digit/Carrier Route
enumeration	A = ADC/AADC
enumeration	M = Mixed ADC & Mixed AADC

simpleType: chargeableEntryLevelType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	B = Destination NDC (Previously BMC)
enumeration	C = Origin SCF
enumeration	D = DDU
enumeration	J = Origin ADC
enumeration	K = Origin NDC (previously BMC)
enumeration	O = Origin Post Office/ DMU
enumeration	R = Destination ADC
enumeration	S = Destination SCF

simpleType: characteristicContentType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	CT = Catalog
enumeration	RR = Reply Envelope or Reply Card
enumeration	NF = Contents NOT required to be mailed FCM
enumeration	DC = DVD/CD or other disk
enumeration	PS = Product Sample
enumeration	RT = Round Trip ONLY: One DVD/CD or other disk (can be LT or FL)
enumeration	MI = Mobile Interactive Technology
enumeration	NP = PACT act

simpleType: characteristicFeeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	OS = Operational Surcharge
enumeration	GS = Geographic Surcharge
enumeration	PP = Picture Permit Indicia
enumeration	MT = Mailer Type

simpleType: characteristicIncentiveType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	MT = Mailer Type
enumeration	VV = Volume w/% off
enumeration	VP = Volume w/price off
enumeration	SB = Small Bus w/% off
enumeration	RE = Revenue
enumeration	RE = Revenue

Tag	Mail.XML 11.0D
enumeration	PI = Piece Count
enumeration	SB = Small Bus w/% off

simpleType: classNotificationType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = (Reserved for future use)
enumeration	B = First-Class Mail
enumeration	C = Periodicals, Initial Notification
enumeration	D = Standard Mail
enumeration	E = Package Services
enumeration	F = Periodicals, Follow-up Notifications

simpleType: closeoutIrregularitiesType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	B = Broken Pallet
enumeration	C = Count don't Match 8125
enumeration	D = Damaged Mail
enumeration	DR = Driver Refused to Assist
enumeration	G = Good
enumeration	IA = Improper Appointment Type
enumeration	IM = Improper Mail Class
enumeration	I = Improper Mail Makeup
enumeration	IF = Incorrect Mail Facility
enumeration	L = Load Unsafe
enumeration	M = Mailing not separated by 8125
enumeration	MI = Miscellaneous
enumeration	NP = No Paperwork (Missing Paperwork)
enumeration	O = Others
enumeration	OW = Overweight Pallets
enumeration	NM = Package on NDC non-machinable
enumeration	P = Pallet Too Tall
enumeration	VS = Vehicle Size
enumeration	CP = Courtesy Pallets
enumeration	CM = Container do not match 8017
enumeration	SM = Separation do not match 8107
enumeration	AR = Altered Required Field on Form
enumeration	IO = Incorrect Entry Office
enumeration	MN = Missing Appointment Number
enumeration	MS = Missing Origin Signature and/or Round Stamp
enumeration	MF = Missing Required Mailer Field on Form
enumeration	MA = Multiple Appts Associated to Load

Tag	Mail.XML 11.0D
enumeration	NF = No Form with Shipment
enumeration	IP = Incorrect Processing Category
enumeration	PD = Past in Home Date

simpleType: closeoutResolutionType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	DCD = Driver Contracted Dispatch Office
enumeration	DCO = Destination Contacted Origin
enumeration	DRL = Driver Reconciled Load
enumeration	INR = Irregularities Count Not be Resolved on Dock

simpleType: communicationRoleType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	From
enumeration	To
enumeration	CC

simpleType: componentIDType

Tag	Mail.XML 11.0D
Base	Removed
maxLength	
minLength	
whiteSpace	

simpleType: componentWeightSourceType

Tag	Mail.XML 11.0D
Base	String
enumeration	A
enumeration	C
enumeration	P
enumeration	L

simpleType: componentWeightStatusType

Tag	Mail.XML 11.0D
Base	String
enumeration	N
enumeration	P
enumeration	F
enumeration	M

simpleType: consigneeContentIDType

Tag	Mail.XML 11.0D
Base	mailxml_base:s09

simpleType: containerBundleStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	-1 = Failed
enumeration	0 = Pending
enumeration	1 = Success

simpleType: containerAcceptanceDateType

Tag	Mail.XML 11.0D
Base	xs:date

simpleType: cccLevelType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	3 = 3-Digit/SCF
enumeration	5 = 5-Digit/Carrier Route
enumeration	A = ADC/AADC
enumeration	M = Mixed ADC & Mixed AADC

simpleType: containerGrossWeightType

Tag	Mail.XML 11.0D
Base	mailxml_base:s05

simpleType: containerHeightType

Tag	Mail.XML 11.0D
Base	mailxml_base:n03

simpleType: containerLevelType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = CR-Direct Eligibility Type = S, T, P
enumeration	B = Mixed CR in 5 Digit Eligibility Type = S, T, P
enumeration	C = Mixed CR in 3 Digit Eligibility Type = S, T
enumeration	D = CR – 5D Scheme Eligibility Type = S, T, P
enumeration	E = FSS Scheme
enumeration	F = FSS Facility

Tag	Mail.XML 11.0D
enumeration	G = 5 Digit (Auto/Presort) Eligibility Type = S, T, P
enumeration	H = 5 Digit (Merged) Eligibility Type = S, T, P
enumeration	I = 5 Digit (Presort Only) Eligibility Type = S, T, P
enumeration	J = 5 Digit (Barcode Only) Eligibility Type = S, T, P
enumeration	K = Metro Scheme Eligibility Type = P
enumeration	M = 5D Scheme (Presort) Eligibility Type = S, T, P
enumeration	N = 5D Scheme (Auto/Presort) Eligibility Type = S, T, P
enumeration	P = 5D Scheme (Barcode) Eligibility Type = S, T, P
enumeration	Q = 5D Scheme (Merged) Eligibility Type = S, T, P
enumeration	R = 3 Digit (Auto, Presort) Eligibility Type = S, T
enumeration	S = 3 Digit (Barcode) Eligibility Type = S, T
enumeration	T = 3 Digit (Presort) Eligibility Type = S, T
enumeration	U = 3 Digit (CR, Auto, Presort) Eligibility Type = S, T, P
enumeration	V = 3 Digit Scheme Eligibility Type = T
enumeration	W
enumeration	X = SCF Eligibility Type = S, P
enumeration	Y = Protected SCF Eligibility Type = P
enumeration	Z = ADC Eligibility Type = S, T, P
enumeration	AA = AADC Eligibility Type = T
enumeration	AB = Mixed ADC Eligibility Type = S, T, P
enumeration	AC = Mixed AADC Eligibility Type = T
enumeration	AD = ASF Eligibility Type = S, P
enumeration	AE = NDC (previously known as BMC) Eligibility Type = S, P
enumeration	AF = Protected NDC Eligibility Type = P
enumeration	AG = Mixed NDC Eligibility Type = S, P
enumeration	AH = Origin MxADC

Tag	Mail.XML 11.0D
enumeration	AI = Protected ADC
enumeration	AJ = Single Piece Container Eligibility Type = S, T
enumeration	AK = MXDS – Mixed Surface Eligibility Type = P
enumeration	AL = MXDA – Mixed Air CSA FCM Only
enumeration	AM = Working Eligibility Type = P
enumeration	AN = Single Piece – First Class Eligibility Type = P
enumeration	AO = Surface Eligibility Type = P
enumeration	AP = Air Eligibility Type = P
enumeration	AQ = Local Eligibility Type = P
enumeration	AR = Origin SCF
enumeration	AS = Origin Mixed ADC Surface
enumeration	AT = Mixed ADC Air
enumeration	BA = Urban Direct Eligibility Type = S, T
enumeration	BB = Rural Direct Eligibility Type = S, T
enumeration	BC = Station Eligibility Type = S, T, P
enumeration	BD = City Eligibility Type = S, T, P
enumeration	BE = FSA Eligibility Type = P
enumeration	BF = DCF Eligibility Type = S, T, P
enumeration	BG = FCP Eligibility Type = S, T, P
enumeration	BH = Province Eligibility Type = P
enumeration	BI = Residual Eligibility Type = S, T, P
enumeration	BJ = Foreign Eligibility Type = S, T, P
enumeration	BK = Country Eligibility Type = S, T, P
enumeration	BL = Mixed Country Eligibility Type = S, T, P
enumeration	BM = M Bags Eligibility Type = S

simpleType: containerPickupDateType

Tag	Mail.XML 11.0D
Base	xs:date

simpleType: containerPickupTimeType

Tag	Mail.XML 11.0D
Base	xs:time

simpleType: containerShipDateType

Tag	Mail.XML 11.0D
Base	xs:date

simpleType: containerShipTimeType

Tag	Mail.XML 11.0D
Base	xs:time

simpleType: contactType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	AccountManager
enumeration	Carrier
enumeration	CrossDock
enumeration	CustomerService
enumeration	HelpDesk
enumeration	DropShipCoordinator
enumeration	ApptCoordinator
enumeration	Plant
enumeration	InboundFreightCoordinator
enumeration	SalesOffice
enumeration	TruckDriver
enumeration	Warehouse
enumeration	Other

simpleType: containerType

Tag	Mail.XML 11.0D
Base	Removed
enumeration	
enumeration	
enumeration	
enumeration	

simpleType: containerStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Accepted
enumeration	OnHold

Tag	Mail.XML 11.0D
enumeration	Rejected

simpleType: containerHeightSourceType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = Actual
enumeration	E = Estimated

simpleType: CQTdatabaseIDType

Tag	Mail.XML 11.0D
Base	xs:nonNegativeInteger
totalDigits	8
whiteSpace	collapse

simpleType: creditDebitIndicator

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	C = Credit
enumeration	D = Debit

simpleType: CRIDType

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	15
minLength	1
pattern	([0-9])*

simpleType: cRIDStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	New
enumeration	Existing

simpleType: csaTemplateType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	PVDS-PER
enumeration	DMU-C
enumeration	DMU-P
enumeration	BMEU
enumeration	UPVDS-PER
enumeration	UDMU-C
enumeration	UDMU-P
enumeration	UBMEU
enumeration	U

simpleType: csaStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	ACT
enumeration	PEN
enumeration	INACT
enumeration	INCOMP
enumeration	PREAPP

simpleType: databaseContainerIDType

Tag	Mail.XML 11.0D
Base	xs:nonNegativeInteger
totalDigits	6

simpleType: dayFrequencyType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Daily
enumeration	Monday
enumeration	Tuesday
enumeration	Wednesday
enumeration	Thursday
enumeration	Friday
enumeration	Saturday
enumeration	Sunday

simpleType: daysOfWeekType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	MON
enumeration	TUE
enumeration	WED
enumeration	THU
enumeration	FRI
enumeration	SAT
enumeration	SUN
enumeration	MF
enumeration	MS
enumeration	ALL
enumeration	H

simpleType: DirectOrMixedCategoryType

Tag	Mail.XML 11.0D
Base	String
enumeration	Direct
enumeration	Mixed

simpleType: dec18-8

Tag	Mail.XML 11.0D
Base	xs:decimal
totalDigits	18
fractionDigits	8

simpleType: deliverabilityCodeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	K = COA Match, No Forwarding Address
enumeration	G = COA Match, PO Box Closed, No Forwarding Address
enumeration	W = COA Match with a temporary COA

simpleType: electronicFileType

Tag	Mail.XML 11.0D
Base	String 1
enumeration	1=Postage & Tracking File
enumeration	2=Tracking File
enumeration	3=Return Service
enumeration	4=Corrections
enumeration	5=Electronic Verification
enumeration	C=Parcel Return Service

simpleType: emailType

Tag	Mail.XML 11.0D
Base	xs:string

simpleType: enclosedByType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	SM = Standard Mail
enumeration	BP = Bound Printed Matter
enumeration	PE = Periodicals
enumeration	PP = Parcel Post
enumeration	ML = Media/Library Mail

simpleType: entryDiscountType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	N = None
enumeration	B = DNDC (previous DBMC)
enumeration	S = DSCF

Tag	Mail.XML 11.0D
enumeration	D = DDU
enumeration	A = DADC
enumeration	C = Origin SCF
enumeration	K = Origin NDC

simpleType: entryFacilityType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	O = Origin
enumeration	B = NDC (Previous BMC)
enumeration	S = SCF
enumeration	D = DDU
enumeration	A = ADC

simpleType: entryPointEDPostalCodeType

Tag	Mail.XML 11.0D
Base	mailxml_base:s06

simpleType: entryPointEDFacilityTypeType

Tag	Mail.XML 11.0D
Base	mailxml_base:s01

simpleType: entryPointADLocaleKeyType

Tag	Mail.XML 11.0D
Base	mailxml_base:s09

simpleType: entryPointADPostalCodeType

Tag	Mail.XML 11.0D
Base	mailxml_base:s09

simpleType: es3

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	3
minLength	3
whiteSpace	preserve

simpleType: es4

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	4
minLength	4
whiteSpace	preserve

simpleType: es5

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	5
minLength	5
whiteSpace	preserve

simpleType: exemptionCodeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	APO
enumeration	APS
enumeration	CAPC
enumeration	DUE
enumeration	HC

simpleType: fASTSchedulerIDType

Tag	Mail.XML 11.0D
Base	mailxml_base:s12

simpleType: feeNotificationType

Tag	Mail.XML 11.0D
Base	String
enumeration	E = Electronic Fee Charged
enumeration	1 = Automated Fee 1st Notice
enumeration	2 = Automated Fee 2nd Notice
enumeration	3 = Automated Fee Charged for 3 or more notices

simpleType: flatRateIndicatorType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	B = Box – Priority Mail Flat Rate Only
enumeration	C = Priority Mail Critical Mail – Letter
enumeration	E = Envelope – Priority Mail Flat Rate Only
enumeration	F = Priority Mail Critical Mail – Flat
enumeration	L = Box – Large Priority Mail Flat Rate Only
enumeration	M = Box – Large Military Destination Priority Mail Flat Rate Only
enumeration	O = Priority Mail Padded Flat Rate Envelope
enumeration	P = Priority Mail Legal Flat Rate Envelope
enumeration	S = Small – Priority Mail Flat Rate Only

simpleType: formType

Tag	Mail.XML 11.0D
Base	xs:string

Tag	Mail.XML 11.0D
enumeration	PM = Priority Mail
enumeration	FC = First-Class
enumeration	SM = Standard Mail
enumeration	BP = Bound Printed Matters
enumeration	PE = Periodicals
enumeration	PP = Parcel Post
enumeration	ML = Media/Library Mail
enumeration	IM = International Mailing

simpleType: fullServiceComplianceIndicatorType

Tag	Mail.XML 11.0D
Base	mailxml_base:yesNo
enumeration	
enumeration	

simpleType: fspiType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	M = Mixed
enumeration	F = Full-Service

simpleType: gPSSystemType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	ECEF
enumeration	HAE
enumeration	MSL

simpleType: IDType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	DriversLicense
enumeration	EmployeeID

simpleType: iMBType

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	34
minLength	20
pattern	([0-9])+

simpleType: IMcbType

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	24
minLength	21

simpleType: IMpbType

Tag	Mail.XML 11.0D
Base	String
maxLength	34
minLength	34

simpleType: iMBarcodeIDType

Tag	Mail.XML 11.0D
Base	mailxml_base:ns02

simpleType: iMBSerialNumber6Type

Tag	Mail.XML 11.0D
Base	mailxml_base:ns06

simpleType: iMBSerialNumber9Type

Tag	Mail.XML 11.0D
Base	mailxml_base:ns09

simpleType: iMBUpperSerializationType

Tag	Mail.XML 11.0D
Base	mailxml_base:s18

simpleType: iMBLowerSerializationType

Tag	Mail.XML 11.0D
Base	mailxml_base:s18

simpleType: IMtbType

Tag	Mail.XML 11.0D
Base	String
maxLength	24
minLength	24

simpleType: includedInOtherDocType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	L = Linked
enumeration	O = Original

simpleType: internalDateType

Tag	Mail.XML 11.0D
Base	xs:date

simpleType: internationalPSPartType

Tag	Mail.XML 11.0D
Base	String
enumeration	A = First Class ail International Permit Imprint
enumeration	B = International Surface Airlift – ISAL
enumeration	C = International Priority Airmail – IPA
enumeration	D = Priority Mail International Permit Imprint
enumeration	E = Global Shipping Software (GSS) Permit Imprint – Priority Mail International
enumeration	F = Global Shipping Software Permit Imprint – Express Mail International
enumeration	G = Global Shipping Software Permit Imprint – Global Express Guaranteed
enumeration	H = Global Bulk Economy Mail Permit Imprint
enumeration	I = Global Direct Canada Admail Permit Imprint
enumeration	J = Global Direct Canada Publication Mail – Permit Imprint
enumeration	K = Global Direct Canada Letter Mail – Permit Imprint
enumeration	Q = Global Direct Permit Imprint
enumeration	R = Reserved
enumeration	S = Extra Services

simpleType: internationalZoneType

Tag	Mail.XML 11.0D
Base	String
enumeration	1 through 15

simpleType: internationalPSPartSectionType

Tag	Mail.XML 11.0D

enumeration	ICLM = Canada Incentive Letter Mail Standard (S/L) – Machinable
	ICLO = Canada Letter Mail Other (Non-Standard and Oversize)
	ICLP = Canada Incentive Letter Mail Standard (S/L) – Presort
	ICLPL = Canada Incentive Letter Mail Standard (S/L) – Presort Local
	ICLS = Canada Letter Mail Standard (S/L)
	ICOM = Canada Incentive Letter Mail Oversize (O/S) – Machinable
	ICOP = Canada Incentive Letter Mail Oversize (O/S) – Presort

Tag	Mail.XML 11.0D

	IEM = Express Mail International
	IEMF = Express Mail International Flat Rate Envelop
	IFCP = LCP Residual/Forward Consolidation Point (FCP)
	IFE = Flat Rate Envelop (Individual Weight Limit: 4 lbs)
	IFLB = Large Flat Rate Box (Individual Weight Limit: 20 lbs)
	IFMB = Medium Flat Rate Box (Individual Weight Limit: 20 lbs)
	IFSB = Small Flat Rate Box (Individual Weight Limit: 4 lbs)
	IGBB = Global Bulk Economy Mail-Base Price-New Jersey Entry
	IGBM = Global Bulk Economy Mail M-Bag New Jersey Entry
	IGD = Global Direct
	IGXG = Global Express Guaranteed (GXG)
	ILDC = LCP Distribution Center Facility
	ILDD = LCP Delivery Mode Direct
	ILDF = LCP Delivery Facility
	IMBG = M-Bag Airmail Sacks
	INDG = NDG Presort
	IPM = Priority Mail International
	IPMI = International Priority Mail IPA
	IPMM = International Priority Mail M-Bag IPA
	IRV = Reserved
	ISAL = International Surface Airlift – ISAL
	ISAM = International Surface Airlift M-Bag – ISAL

simpleType: internationalPriceGroupType

Tag	Mail.XML 11.0D
Base	String
enumeration	1 = Canada
enumeration	2 = Mexico
enumeration	3 through 5 = China, Eastern Europe, Hong Kong, Japan, South Korea, Turkey, Western Europe
enumeration	6 through 9 = Africa, Asia (exception include price group 3 through 5), central America, Middle East, South America
enumeration	10 through 15 = Unknown – for future use

simpleType: inCountyOutCountyIndicatorType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	I – In County
enumeration	N = Outside County
enumeration	

simpleType: jobIDType

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	8
minLength	1
whiteSpace	Preserve

simpleType: localeKeyType

Tag	Mail.XML 11.0D
Base	mailxml_base:s09

simpleType: inductionStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = Accepted
enumeration	N = Rejected

simpleType: logicalPhysicalBundleIndicatorType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	P = Physical
enumeration	L = Logical

simpleType: logicalPhysicalContainerIndicatorType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	P = Physical

Tag	Mail.XML 11.0D
enumeration	L = Logical

simpleType: mailAgreementTypeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = Alternate Mailing System
enumeration	B = Optional Procedure
enumeration	C = Manifest Mailing
enumeration	D = Value Added
enumeration	E = Combined Mail
enumeration	F = Combined and Value Added

simpleType: mailClassType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	1 = First-Class Mail
enumeration	2 = Periodicals
enumeration	3 = Standard Mail
enumeration	4 = Package Services
enumeration	6 = (Future Placeholder)

simpleType: maildatContainerStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	R = Ready to Pay
enumeration	X = Paid
enumeration	C = Cancel
enumeration	D = Delete
enumeration	P = Preliminary Postage Statement
enumeration	T = Transportation Information Update

simpleType: mailerID6Type

Tag	Mail.XML 11.0D
Base	mailxml_base:ns06

simpleType: mailerID9Type

Tag	Mail.XML 11.0D
Base	mailxml_base:ns09

simpleType: mailXMLContainerType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	1 = #1 Sack
enumeration	2 = #2 Sack
enumeration	3 = #3 Sack

Tag	Mail.XML 11.0D
enumeration	4 = 01V Sack
enumeration	5 = 03V Sack
enumeration	6 = OD Sack
enumeration	7 = OD Parcel
enumeration	8 = OD Full Tray Box
enumeration	9 = OD Half Tray Box
enumeration	10 = Flat Trays
enumeration	11 = FRE/FRB
enumeration	12 = OD Surface Transported Pallets
enumeration	13 = Full Gaylords
enumeration	14 = Half(1/2) Gaylords
enumeration	E = EMM Tray
enumeration	F = Flat Tub
enumeration	L = Logical Tray
enumeration	O = 1' Tray
enumeration	P = Pallet
enumeration	S = Sack (General)
enumeration	T = 2' Tray
enumeration	V = Sack (Virtual)
enumeration	B = Bedload
enumeration	U – Unit Load Device
enumeration	Z = User Pallet
enumeration	X = (Future Placeholder)
enumeration	H = EIRS 61 – Hamper, Large Canvas
enumeration	A = EIRS 61P – Hamper, Large Plastic
enumeration	G = EIRS 66 – General Purpose Mail Container w/Gate
enumeration	D = EIRS 68 – Eastern Region Mail Container w/Web Door
enumeration	R = EIRS 84 – Wire Container Rigid
enumeration	C = EIRS 84C – Collapsible Wire Container
enumeration	M = Logical Pallet (MLOCR)
enumeration	AB = Air Box *Note: For Mail.XML 11.0D when this value is provided – the PostalOne! system will reject the message with appropriate message.*
enumeration	Y = PMOD Sack

simpleType: mailXMLPresentationCategoryType

Tag	Mail.XML 11.0D
Base	xs:string

Tag	Mail.XML 11.0D
enumeration	M = MLOCR
enumeration	P = Conventional Presort
enumeration	I = Manifest Individual
enumeration	N = Single Piece
enumeration	S = Manifest Summary

simpleType: manifestType

Tag	Mail.XML 11.0D
enumeration	DM = Domestic Mail
enumeration	IM = International Mail

simpleType: manifestMailClassType

Tag	Mail.XML 11.0D
enumeration	BB = BoundPrinted Material
enumeration	BL = Library Rate
enumeration	BP = Parcel Post
enumeration	BS = Media Mail
enumeration	CP = Priority Mail International Parcels
enumeration	DM = Domestic Matter for the Blind
enumeration	EX = Express Mail®
enumeration	FC = First Class Mail®
enumeration	IE = Express Mail International
enumeration	IL = Priority Mail International Letters
enumeration	LC = First Class Mail International
enumeration	PG = Global Express Guaranteed
enumeration	PM = Priority Mail
enumeration	S2 = Standard Mail Non-Profit
enumeration	S3 = Standard Mail Enhanced Carrier Route
enumeration	S4 = Standard Mail Nonprofit Enhance Carrier Route
enumeration	SA = Standard Mail

simpleType: manifestRateLevelType

Tag	Mail.XML 11.0D

enumeration	AD = "ADC Rate”
	BA = "Basic Rate"
	BM = "Balloon Rate"
	BP = "Basic Bound Printed Matter Pre-bar coded"
	BS = "NDC Special Standard"
	B3 = "3-Digit Balloon Rate"
	CB = "Carrier Route Basic Rate"
	CD = "Express Mail Custom

Tag	Mail.XML 11.0D

	Design"
	CH = "Carrier Route High Density Rate"
	CR = "Carrier Route Rate"
	CS = "Carrier Route Saturation Rate"
	C1 = "Cubic Pricing Tier 1"
	C2 = "Cubic Pricing Tier 2"
	C3 = "Cubic Pricing Tier 3"
	C4 = "Cubic Pricing Tier 4"
	C5 = "Cubic Pricing Tier 5"
	DN = "Dimensional Nonrectangular Rate"
	DR = "Dimensional Rectangular Rate"
	E1 = "Express Mail Same Day Airport Flat Rate"
	E3 = "Express Mail Flat Rate Hold For Pickup"
	E4 = "Express Mail Flat Rate Post Office To Addressee"
	FB = "Priority Mail Medium Flat Rate Box"
	FE = "Flat Rate Envelope"
	IG = "P/PS<15Lbs, combined length/girth >84" & <= 108" subject to balloon rate"
	IM = "P/ PS between 108" and 130" subject to oversize rate"
	MA = "Mixed ADC Rate"
	MB = "Mixed NDC Rate"
	NP = "Non-presorted Rate"
	OS = "Oversize"
	PG = "PM<15lbs combined length/girth >84" & <=108" subject to balloon rate"
	PL = "Priority Mail Large Flat Rate Box"
	PM = "Priority Mail Large Flat Rate Box APO/FPO"
	PP = "Express Mail Hold For Pickup"

Tag	Mail.XML 11.0D
	PR = "Presorted Rate" RB = "Bulk Basic" SB = "Media Mail NDC" SP = "Single Piece Rate" S1 = "Single Piece – Flat" S2 = "Single Piece – Parcel" S5 = "Media Mail 5 Digit" 1A = "Intra-NDC Single Piece Rate" 1E = "Inter-NDC Single Piece Rate" 2A = "Intra-NDC Balloon Rate" 2E = "Inter-NDC Balloon Rate" 3A = "Intra-NDC Oversized Rate" 3D = "3-Digit Rate" 3E = "Inter-NDC Oversized Rate" 5D = "5 Digit Rate" 5S = "5 Digit Special Standard"

simpleType: manifestExtraServiceCode

Tag	Mail.XML 11.0D
enumeration	420 = Parcel Return Service
enumeration	430 = Open & Distribute
enumeration	440 – Re Shipment
enumeration	910 = Certified Mail
enumeration	915 = COD
enumeration	920 – Delivery Confirmation
enumeration	921 = Signature Confirmation
enumeration	925 = Express Mail Merchandise Insurance
enumeration	930 = Insurance <=$200
enumeration	931 = Insurance >$200
enumeration	935 = Parcel Airlift (PAL)
enumeration	940 = Registered Mail

enumeration	950 = Restricted Delivery
enumeration	955 = Return Receipt
enumeration	957 = Return Receipt Electronic
enumeration	960 = Return Receipt for Merchandise
enumeration	970 = Special Handling
enumeration	980 = Merchandise Return Service (MRS)
enumeration	985 = Hold for Pickup

simpleType: mIDStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	New
enumeration	Existing

simpleType: meteredAffixedMethodType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	C = Metered - Correct
enumeration	L = Metered – Lowest
enumeration	M = Metered - Neither

simpleType: monthType

Tag	Mail.XML 11.0D
Base	xs:nonNegativeInteger
maxInclusive	12
minInclusive	1

simpleType: moveUpdateMethodType

Tag	Mail.XML 11.0D
Base	xs:integer
enumeration	0 = None
enumeration	1 = ACS
enumeration	2 = NCOA
enumeration	3 = FAST Forward
enumeration	4 = Mailer Move Update Process
enumeration	5 = Ancillary Service Endorsement
enumeration	6 = Multiple
enumeration	7 = N/A
enumeration	8 = OneCode ACS

simpleType: MPUIDType

Tag	Mail.XML 11.0D
Base	xs:nonNegativeInteger
totalDigits	5

simpleType: ns02

Tag	Mail.XML 11.0D
Base	xs:string
pattern	[0-9]{2}

simpleType: ns03

Tag	Mail.XML 11.0D
Base	xs:string
pattern	[0-9]{3}

simpleType: ns04

Tag	Mail.XML 11.0D
Base	xs:string
pattern	[0-9]{4}

simpleType: ns05

Tag	Mail.XML 11.0D
Base	xs:string
pattern	[0-9]{5}

simpleType: ns06

Tag	Mail.XML 11.0D
Base	xs:string
pattern	[0-9]{6}

simpleType: ns09

Tag	Mail.XML 11.0D
Base	xs:string
pattern	[0-9]{9}

simpleType: ns10

Tag	Mail.XML 11.0D
Base	xs:string
pattern	[0-9]{10}

simpleType: ns11

Tag	Mail.XML 11.0D
Base	xs:string
pattern	[0-9]{11}

simpleType: ns22

Tag	Mail.XML 11.0D
Base	xs:string
pattern	[0-9]{22}

simpleType: n01

Tag	Mail.XML 11.0D
Base	xs:positiveInteger
totalDigits	1

simpleType: n02

Tag	Mail.XML 11.0D
Base	xs:positiveInteger
totalDigits	2

simpleType: n03

Tag	Mail.XML 11.0D
Base	xs:nonNegativeInteger
totalDigits	3

simpleType: n05

Tag	Mail.XML 11.0D
Base	xs:nonNegativeInteger
totalDigits	5

simpleType: n06

Tag	Mail.XML 11.0D
Base	xs:nonNegativeInteger
totalDigits	6

simpleType: addressType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	F = Foreign
enumeration	G = General Delivery
enumeration	H = Highway Route Contract
enumeration	P = Post Office Box
enumeration	R = Rural Route
enumeration	S = Street Address
enumeration	U = Urbanization Name
enumeration	V = Highway Contract Route with box number
enumeration	X = Rural Route with box number

simpleType: operatingModeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	N = Normal
enumeration	B = ByPass

simpleType: onePassPlannedOrActualFlagType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = Actual
enumeration	P = Planned

simpleType: packageIDType

Tag	Mail.XML 11.0
Base	mailxml_base:s06

simpleType: PackageServicesType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	PS
enumeration	BPM
enumeration	LIB
enumeration	MEDIA
enumeration	PP

simpleType: packageCountRangeType

Tag	Mail.XML 11.0D
Base	xs:Integer
Pattern	[1-9]|[1-9][0]

simpleType: packagingMethodType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	0 = Piece Count
enumeration	1 = Weight
enumeration	2 = Both

simpleType: palletHandlingUnitType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Sacks
enumeration	Bundles
enumeration	MixedLengthTrays
enumeration	1ft Tray
enumeration	2ft Tray
enumeration	EMM Tray
enumeration	Flat Tub
enumeration	Parcels

simpleType: palletType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Pallet
enumeration	EIRS 61 - Hamper, Large Canvas
enumeration	EIRS 61P - Hamper, Large Plastic
enumeration	EIRS 66 - General Purpose Mail Container w/Gate
enumeration	EIRS 68 - Eastern Region Mail Container w/Web Door
enumeration	EIRS 84 - Wire Container Rigid
enumeration	EIRS 84C - Collapsible Wire Container

simpleType: parentContainerRefIDType

Tag	Mail.XML 11.0D
Base	mailxml_base:s06

simpleType: percentageType

Tag	Mail.XML 11.0D
Base	xs:decimal
totalDigits	5
fractionDigits	2
minInclusive	0.00
maxInclusive	100.00

simpleType: periodicalPriceType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	D = Non-Advertising Pound Rate – DSCF (B13)
enumeration	C = Non-advertising Pound Rate – DDU (B12)
enumeration	E = Non-Advertising Pound Rate – DADC (B14)
enumeration	R = Repositionable notes (surcharge) (A21, F2)
enumeration	Z = Non-Advertising Pound Rate – All Others (B15)
enumeration	5 = Non-Advertising Piece Rate (C30)
enumeration	9 = Ride-along Surcharge (A20, F1)
enumeration	1 = DDU (A17)

simpleType: permitTypeType

Tag	Mail.XML 11.0
Base	xs:string
enumeration	PI = Permit Imprint
enumeration	MT = Metered Tape
enumeration	PC = Pre-cancelled Stamps
enumeration	BR = Business Reply
enumeration	PE = Periodicals
enumeration	GH = Ghost

Tag	Mail.XML 11.0
enumeration	OI = Official Imprint
enumeration	OM = Official Imprint
enumeration	PP = Pending Periodicals
enumeration	PA = Partial Permit Imprint

simpleType: phoneNumberType

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	25
minLength	1
whiteSpace	preserve

simpleType: pickupOrDeliveryType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Pickup
enumeration	Delivery

simpleType: planetCodeType

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	13
minLength	11
pattern	[0-9]{2}

simpleType: postageGroupingIDType

Tag	Mail.XML 11.0D
Base	mailxml_base:s08

simpleType: postagePaymentMethodType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	S
enumeration	M
enumeration	P

simpleType: postageStatementLineNumber1Type

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	E26
enumeration	E27
enumeration	E28
enumeration	E29
enumeration	E30
enumeration	E31
enumeration	E32

Tag	Mail.XML 11.0D
enumeration	E33
enumeration	E34
enumeration	E35
enumeration	E36
enumeration	E37
enumeration	E38
enumeration	E39
enumeration	E40
enumeration	E41
enumeration	E42
enumeration	E43
enumeration	E44
enumeration	E45
enumeration	E46

simpleType: postageStatementLineNumber2Type

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	E1
enumeration	E2
enumeration	E3
enumeration	E4
enumeration	E5
enumeration	E6
enumeration	E7
enumeration	E8
enumeration	E9
enumeration	E10
enumeration	E11
enumeration	E12
enumeration	E13
enumeration	E14
enumeration	E15
enumeration	E16
enumeration	E17
enumeration	E18
enumeration	E19
enumeration	E20
enumeration	E21
enumeration	E22
enumeration	E23
enumeration	E24

simpleType: postageStatementLineNumber3Type

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	D1
enumeration	D2

Tag	Mail.XML 11.0D
enumeration	D3
enumeration	D4
enumeration	D5
enumeration	D6
enumeration	D7
enumeration	D8
enumeration	D9
enumeration	D10
enumeration	D11
enumeration	D12
enumeration	D13
enumeration	D14
enumeration	D15
enumeration	D16
enumeration	D17

simpleType: postageStatementStatusType

Tag	Mail.XML 11.0
Base	xs:string
enumeration	UPD = Updated
enumeration	FIN = Finalized
enumeration	REW = Rework
enumeration	REV = Reversed
enumeration	CAN = Cancelled
enumeration	FAI = Failed Verification
enumeration	PEN = Pending
enumeration	PAS = Passed
enumeration	INC = Incomplete
enumeration	NAP = Not Available for Processing
enumeration	EST = Estimate Postage
enumeration	CON = Consolidated Postage Statement
enumeration	ERR = Error
enumeration	COM = Complete
enumeration	AVW = Address Verification Warning
enumeration	PAV = Pending Address Verification
enumeration	FPP = Finalized Pending Payment

simpleType: priceType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	commercial
enumeration	commercialplus
enumeration	retail
enumeration	NSA

simpleType: primarySecondaryIndicatorType

Tag	Mail.XML 11.0D
Base	xs:string

Tag	Mail.XML 11.0D
enumeration	P = Primary
enumeration	S = Secondary

simpleType: processingCategoryType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	LT = Letter
enumeration	FL = Flat
enumeration	CD = Card
enumeration	OS = Outside Parcel
enumeration	MP = Machinable Parcel
enumeration	IR = Irregular Parcel
enumeration	PF = Parcel, First Class
enumeration	CM = Custom Mail
enumeration	NP = Non-Machinable Parcel
enumeration	SP = Small Packets
enumeration	MB = M-Bags

simpleType: processingCodeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	S = Surface
enumeration	MXDS = Mixed Surface
enumeration	L = Locale
enumeration	WKG = Working Mail
enumeration	A = Air Separation
enumeration	MXDA = Mixed Air
enumeration	SP = Single Piece

simpleType: pvdsContainerType

Tag	Mail.XML 11.0D
Base	-
enumeration	

simpleType: pvdsContainerSubType

Tag	Mail.XML 11.0D
Base	Removed
enumeration	
enumeration	

simpleType: qualificationReportClassType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	FC = First Class
enumeration	S = Standard
enumeration	P = Periodicals

Tag	Mail.XML 11.0D
enumeration	PM = Priority Mail
enumeration	PP = Parcel Post
enumeration	A = Air Mail

simpleType: rateCategoryType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = Saturation ECR
Enumeration	A1 = Non-Automation Saturation – ECR
enumeration	AF = Saturation ECT Letter
enumeration	B = High Density ECR
Enumeration	B1 = Non-Automation High Density - ECR
enumeration	BF = High Density SCR Letter
enumeration	D = Carrier Route
enumeration	D1 = Non Automation Carrier Route
enumeration	DF = Carrier Letter
enumeration	E = 5 Digit Barcode
enumeration	FB = Firm Bundle (Not In-County)
enumeration	G = 5D Non-Barcode
enumeration	GF = 5D Letter
enumeration	H = 3 Digit Barcode
enumeration	K = 3D Non-Barcode
enumeration	KF = 3D Letter
enumeration	L = Basic Barcode
enumeration	L1 = AADC Barcode
enumeration	L2 = Mixed AADC Barcode
enumeration	L3 = ADC Barcode
enumeration	L4 = Mixed ADC Barcode
enumeration	L5 = ADC Non-Barcode
enumeration	L6 = Mixed ADC Non-Barcode
enumeration	L7 = AADC Non-Barcode
enumeration	L8 = Mixed AADC Non-Barcode
enumeration	N = Basic Non-Automation (Basic Presorted)
enumeration	O = Carrier Route – Barcode
enumeration	O1 = Non Automation Carrier Route – Barcode * CURRENTLY NOT ALLOWED TO BE USED.
enumeration	PI = 5 Digit Parcel 0.2063 Pounds or less
enumeration	PM = 3 Digit Parcel 0.2063 Pounds or less
enumeration	P7 = NDC Parcel 0.2063 pounds or less
enumeration	P8 = Mixed NDC Parcel 0.2063 Pounds or less
enumeration	PE = 5 Digit Machinable Parcel over 0.2063 Pounds
enumeration	PQ = NDC Machinable Parcel over 0.2063 Pounds
enumeration	PR = Mixed NDC Machinable Parcel over 0.2063 Pounds
enumeration	PG = 5 Digit Irregular Parcel over 0.2063 Pounds
enumeration	PK = 3 Digit Irregular Parcel over 0.2063 Pounds
enumeration	P5 = NDC Irregular Parcel over 0.2063 Pounds

Tag	Mail.XML 11.0D
enumeration	P6 = Mixed NDC Irregular Parcel over 0.2063 Pounds
enumeration	S = Single Piece
enumeration	SD = Single-Piece/Does not meet content standards
enumeration	Z1 = Parcel Post (NDC Sort)
enumeration	Z2 = Parcel Post (ONDC Sort)
enumeration	P10 = Priority Mail Regional Rate Box A (max weight 15 lbs)
enumeration	P11 = Priority Mail Regional Rate Box B (max weight 20 lbs)
enumeration	P12 = Priority Mail – Cibic Tier 1 (up to .10 lbs)
enumeration	P13 = Priority Mail – Cubic Tier 2 (up to .20 lbs)
enumeration	P14 = Priority Mail – Cubic Tier 3 (up to .30 lbs)
enumeration	P15 = Priority Mail – Cubic Tier 4 (up to .40 lbs)
enumeration	P16 = Priority Mail – Cubic Tier 5 (up to .50 lbs)
enumeration	IA10 = International Postcards
enumeration	IA11 = Letters (up to 3.5 ounces), Flats and Packages/Small Packets (up to 4 pounds)
enumeration	IA12 = Nonmachinable Surcharge (letters only, up to 3.5 ounces)
enumeration	IA13 = M-Bag Airmail
enumeration	IB14 = Worldwide Nonpresort
enumeration	IB15 = International Surface Air Lift-M-Bag-ISAL
enumeration	IC15 = International Priority Mail-IPA
enumeration	IC17 = Worldwide Nonpresort
enumeration	IC18 = International Priority Mail-M-Bag-IPA
enumeration	ID19 = Priority Mail International
enumeration	ID20 = Flat Rate Envelopes (Individual Weight Limit: 4 lbs.) (15" x 9 1/2" or Smaller)
enumeration	ID21 = Padded Flat Rate Envelope (Individual Weight Limit: 4 lbs.)
enumeration	ID22 = Small Flat Rate Boxes (Individual Weight Limit: 4 lbs.) (Including items O-Small FRBX, ODVDS, and O-1096-L)
enumeration	ID23 = Medium Flat Rate Boxes (Individual Weight Limit: 20 lbs.)
enumeration	ID24 = Large Flat Rate Box (Individual Weight Limit: 20 lbs.)
enumeration	IE25 = Flat Rate Envelopes (Individual Weight Limit: 4 lbs.) (15" x 9 1/2" or Smaller)
enumeration	IE26 = Padded Flat Rate Envelope (Individual Weight Limit 4 lbs.)
enumeration	IE27 = Small Flat Rate Boxes (Individual Weight Limit: 4 lbs.) (Including items O-Small FRBX, ODVDS, and O-1096-L)
enumeration	IE28 = Medium Flat Rate Boxes (Individual Weight Limit: 20 lbs.)
enumeration	IE29 = Large Flat Rate Box (Individual Weight Limit: 20 lbs.)
enumeration	IF30 = Express Mail International
enumeration	IF31 = Express Mail International Flat Rate Envelopes

Tag	Mail.XML 11.0D
	(15" x 9 1/2" or Smaller)
enumeration	IG32 = Global Express Guaranteed (GXG)
enumeration	IH33 = Global Bulk Economy Mail-Base Prices-New Jersey Entry
enumeration	IH34 = Global Bulk Economy Mail-M-Bag - New Jersey Entry
enumeration	II35 = NDG Presort
enumeration	II36 = LCP Delivery Mode Direct
enumeration	II37 = LCP Delivery Facility
enumeration	II38 = LCP Distribution Center Facility (DCF)
enumeration	II39 = LCP Forward Consolidation Point (FCP)
enumeration	II40 = LCP Residual
enumeration	II41 = Excess Weight Postage-if applicable (for items over 0.1100 lbs. up to 1.100 lbs. [over 1.76 oz. up to 17.6 oz.])
enumeration	IJ42 = NDG Presort
enumeration	IJ43 = LCP Delivery Mode Direct
enumeration	IJ44 = LCP Delivery Facility
enumeration	IJ45 = LCP Distribution Center Facility (DCF)
enumeration	IJ46 = LCP Forward Consolidation Point (FCP)
enumeration	IJ47 = LCP Residual
enumeration	IK48 = Canada Lettermail - Standard (S/L) - up to 1.0576 oz.
enumeration	IK49 = Canada Lettermail - Standard (S/L) -over 1.0576 oz. and up to 1.7632 oz.
enumeration	IK50 = Canada Lettermail - Other Lettermail (Non-standard and Oversize) - up to 3.5264 oz.
enumeration	IK51 = Canada Lettermail - Other Lettermail (Non-standard and Oversize) - over 3.5264 oz. and up to 7.0544 oz.
enumeration	IK52 = Canada Lettermail - Other Lettermail (Non-standard and Oversize) - over 7.0544 oz. and up to 10.5808 oz.
enumeration	IK53 = Canada Lettermail - Other Lettermail (Non-standard and Oversize) - over 10.5808 oz. and up to 14.1088 oz.
enumeration	IK54 = Canada Lettermail - Other Lettermail (Non-standard and Oversize) - over 14.1088 oz. and up to 17.6368 oz.
enumeration	IK55 = Canada Incentive Lettermail - Standard (S/L) Machineable - up to 1.0576 oz.
enumeration	IK56 = Canada Incentive Lettermail - Standard (S/L) Machineable - over 1.0576 oz. and up to 1.7632 oz.
enumeration	IK57 = Canada Incentive Lettermail - Standard (S/L) Machineable - over 1.7632 oz. and up to 3.5264 oz.
enumeration	IK58 = Canada Incentive Lettermail - Standard (S/L) - Presort - up to 1.0576 oz.
enumeration	IK59 = Canada Incentive Lettermail - Standard (S/L) - Presort - over 1.0576 oz. and up to 1.7632 oz.
enumeration	IK60 = Canada Incentive Lettermail - Standard (S/L) -

Tag	Mail.XML 11.0D
	Presort - over 1.7632 oz. and up to 3.5264 oz.
enumeration	IK61 = Canada Incentive Lettermail - Standard (S/L) - Presort Local - up to 1.0576 oz.
enumeration	IK62 = Canada Incentive Lettermail - Standard (S/L) - Presort Local - over 1.0576 oz. and up to 1.7632 oz.
enumeration	IK63 = Canada Incentive Lettermail - Standard (S/L) - Presort Local - over 1.7632 oz. and up to 3.5264 oz.
enumeration	IK64 = Canada Incentive Lettermail - Oversize (O/S) - Machineable - up to 3.5264 oz.
enumeration	IK65 = Canada Incentive Lettermail - Oversize (O/S) - Machineable - over 3.5264 oz. and up to 17.6368 oz.
enumeration	IK66 = Canada Incentive Lettermail - Oversize (O/S) - Presort - up to 3.5264 oz.
enumeration	IK67 = Canada Incentive Lettermail - Oversize (O/S) - Presort - over 3.5264 oz. and up to 17.6368 oz.
enumeration	IQ68 = Global Direct
enumeration	IR69 = Reserved
enumeration	PS1
enumeration	PS2
enumeration	PS3
enumeration	PS4
enumeration	PS5
enumeration	PS6
enumeration	PS7
enumeration	PS8
enumeration	PS9
enumeration	PS10
enumeration	PS11

simpleType: rateLevelType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	SP = Single Piece
enumeration	5B = Auto 5 Digit
enumeration	3B = Auto 3 Digit
enumeration	AB = Auto ADC/AADC
enumeration	PSRT = Presort
enumeration	5D = Presort 5 Digit
enumeration	3D = Presort 3 Digit
enumeration	NDC = NDC Presort
enumeration	ONDC = ONDC Presort
enumeration	BP = Standard Mail NDC
enumeration	MP = Mixed NDC Standard
enumeration	AD = ADC Presort
enumeration	MD = Mixed ADC Presort
enumeration	WS = Carrier Route Saturation
enumeration	HD = Carrier Route High Density
enumeration	CR = Carrier Route Basic
enumeration	MB = Auto Mixed ADC/AADC

Tag	Mail.XML 11.0D
enumeration	FB = Firm Bundle
enumeration	BB = Barcode Basic
enumeration	SCF

simpleType: rateLevelIndicatorType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = Auto (use with Rate Level 5B, 3B, AB, MB)
enumeration	B = Non-Auto (use with Rate Level 5D, 3D, AD, MD)
enumeration	C = ECR (use with Rate Level WS, HD, CR)
enumeration	M = Machinable (use with Rate Level AB, MB)
enumeration	N = Non-Machinable (use with Rate Level 5D, 3D, AD, MD)

simpleType: rateType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	R = Regular
enumeration	L = Library Mail
enumeration	B = Bound Printed Matter
enumeration	C = Classroom
enumeration	S = Science of Agriculture
enumeration	F = Media Mail
enumeration	A = Alt Delivery
enumeration	N = Non-Profit
enumeration	P = Parcel Post
enumeration	T = Priority Mail
enumeration	X = Other
enumeration	H = (Placeholders)
enumeration	M = (Placeholders)
enumeration	Z = (Placeholders)
enumeration	E = Priority Mail Flat Fixed (Rate Envelope)
enumeration	G = Priority Mail Flat Fixed (rate Box)
enumeration	1 = UA (International)
enumeration	2 = UL (International)
enumeration	3 = UM (International)
enumeration	4 = UR (International)
enumeration	J = Priority Mail Flat – Large Box
enumeration	K = Priority Mail Flat – Large Box APO/FPO
enumeration	V=FCM Election Mail
enumeration	W = Science and Agriculture Limited
enumeration	Y = Regular Limited Circulation

simpleType: reasonCodeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = Attempted – Not Known
enumeration	B = Returned for Better Address
enumeration	D = Outside Delivery Limits
enumeration	E = In Dispute
enumeration	I = Insufficient Address
enumeration	L = Illegible
enumeration	M = No Mail Receptacle
enumeration	N = No such number
enumeration	P = Deceased
enumeration	Q = Not Deliverable as Addressed/ Unable to Forward
enumeration	R = Refused
enumeration	S = No Such Street
enumeration	U = Unclaimed
enumeration	V = Vacant
enumeration	X = No Such Office

simpleType: recurringStatusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Active
enumeration	Pending
enumeration	All

simpleType: reservationNumberType

Tag	Mail.XML 11.0D
Base	mailxml_base:s15

simpleType: roleType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Data Update Agent
enumeration	Owner
enumeration	Confirm Subscriber
enumeration	Confirm Data Receiver
enumeration	ACS Data Receiver
enumeration	ACS Subscriber
enumeration	List Owner
enumeration	Mail Creator
enumeration	Addressor
enumeration	Comail
enumeration	CoMingler
enumeration	MLOCR
enumeration	Mail Owner
enumeration	Mail Agent

Tag	Mail.XML 11.0D
enumeration	Entry Point Planner
enumeration	Shipment Planner
enumeration	Consolidator
enumeration	Secondary Consolidator
enumeration	Appointment Management Receiver
enumeration	Consignee/Product Receiver
enumeration	Dispatcher
enumeration	Tracker CONFIRM-scan
enumeration	Tracker CONFIRM-payer
enumeration	Tracker Seed Source
enumeration	Tracker Piece-level-scan
enumeration	Tracker ASN-scan
enumeration	Tracker Induction-scan
enumeration	Tracker Planet-code
enumeration	ACS receiver
enumeration	ACS payer
enumeration	Seamless Acceptance
enumeration	Seamless Verification
enumeration	Security
enumeration	Logistics Provider
enumeration	Carrier
enumeration	Data Brokering Agent
enumeration	Nixie
enumeration	StartTheClock
enumeration	ContainerVisibility
enumeration	DeliveryPointValidation

simpleType: responsibleTransportationPartyType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Shipper
enumeration	Consignee

simpleType: RoutingZipCodeType

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	11
minLength	0

simpleType: s01

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	1
minLength	1
whiteSpace	preserve

simpleType: s02

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	2
minLength	1
whiteSpace	preserve

simpleType: s03

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	3
minLength	1
whiteSpace	preserve

simpleType: s04

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	4
minLength	1
whiteSpace	preserve

simpleType: s05

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	5
minLength	1
whiteSpace	preserve

simpleType: s06

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	6
minLength	1
whiteSpace	preserve

simpleType: s07

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	7
minLength	1
whiteSpace	preserve

simpleType: s08

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	8

Tag	Mail.XML 11.0D
minLength	1
whiteSpace	preserve

simpleType: s09

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	9
minLength	1
whiteSpace	preserve

simpleType: s10

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	10
minLength	1
whiteSpace	preserve

simpleType: s11

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	11
minLength	1
whiteSpace	preserve

simpleType: s12

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	12
minLength	1
whiteSpace	preserve

simpleType: s15

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	15
minLength	1
whiteSpace	preserve

simpleType: s16

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	16
minLength	1
whiteSpace	preserve

simpleType: s18

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	18
minLength	1
whiteSpace	preserve

simpleType: s20

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	20
minLength	1
whiteSpace	preserve

simpleType: s22

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	22
minLength	1
whiteSpace	preserve

simpleType: s25

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	25
minLength	1
whiteSpace	preserve

simpleType: s28

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	28
minLength	1
whiteSpace	preserve

simpleType: s30

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	30
minLength	1
whiteSpace	preserve

simpleType: s31

Tag	Mail.XML 11.0D
Base	Removed

Tag	Mail.XML 11.0D
maxLength	
minLength	
whiteSpace	

simpleType: s37

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	37
minLength	1
whiteSpace	preserve

simpleType: s40

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	40
minLength	1
whiteSpace	preserve

simpleType: s45

Base	xs:string
maxLength	45
minLength	1
whiteSpace	preserve
Tag	Mail.XML 11.0D

simpleType: s50

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	50
minLength	1
whiteSpace	Preserve

simpleType: s60

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	60
minLength	1
whiteSpace	Preserve

simpleType: s64

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	64
minLength	1

Tag	Mail.XML 11.0D
whiteSpace	Preserve

simpleType: s66

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	66
minLength	1
whiteSpace	Preserve
Tag	Mail.XML 11.0D

simpleType: s80

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	80
minLength	1
whiteSpace	Preserve

simpleType: s256

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	256
minLength	1
whiteSpace	Preserve

simpleType: s260

Base	xs:string
maxLength	260
minLength	1
whiteSpace	Preserve

simpleType: sackingBasisType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	0 = Piece Count
enumeration	1 = 20 lbs.
enumeration	2 = 125 Pieces
enumeration	3 = 15 lbs.
enumeration	4 = Both

simpleType: scheduledInductionDateType

Tag	Mail.XML 11.0D
Base	xs:date

simpleType: scheduledInductionTimeType

Tag	Mail.XML 11.0D

Tag	Mail.XML 11.0D
Base	xs:time

simpleType: scheduledInHomeDateType

Tag	Mail.XML 11.0D
Base	xs:date

simpleType: serialNumberSevenOrEleven

Tag	Mail.XML 11.0D
Base	String
Pattern	"[0-9]{7}|[0-9]{11}"

simpleType: serialNumberTenOrFourteen

Tag	Mail.XML 11.0D
Base	String
Pattern	"[0-9]{10}|[0-9]{14}"

simpleType: serviceLevelIndicatorType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	F = Full-Service
enumeration	B = Basic
enumeration	P = POSTNET
enumeration	O = Other

simpleType: serviceCodeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	CM = Certified Mail
enumeration	COD = Collect on Delivery
enumeration	COM = Certificate of Mailing
enumeration	DC = Delivery Confirmation
enumeration	IM = Insured Mail
enumeration	PAL = Parcel Airlift
enumeration	RD = Restricted Delivery
enumeration	RM = Registered Mail
enumeration	RR = Return Receipt
enumeration	RRE = Return Receipt Electronic
enumeration	RRM = Return Receipt for Merchandise
enumeration	SC = Signature Confirmation
enumeration	SH = Special Handling
enumeration	AS = Adult Signature
enumeration	RAS = Restricted Adult Signature
enumeration	DCD=Day Certain Deliery

simpleType: ServiceTypeCodeType

Tag	Mail.XML 11.0D
Base	mailxml_base:ns03

simpleType: sortationLevelType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	CRD
enumeration	5DG
enumeration	CR5
enumeration	5DGS
enumeration	CR5S
enumeration	M5D
enumeration	M5DS
enumeration	MET
enumeration	3DG
enumeration	CR3
enumeration	3DGS
enumeration	ADC
enumeration	AADC
enumeration	MADC
enumeration	MAAD
enumeration	SCF
enumeration	PSCF
enumeration	NDC
enumeration	PNDC
enumeration	MNDC
enumeration	ASF
enumeration	OMX
enumeration	PADC
enumeration	MRDC
enumeration	SP
enumeration	MXDS
enumeration	MXDA
enumeration	WORKING
enumeration	SINGLEPIECE
enumeration	SURFACE
enumeration	AIR
enumeration	LOCAL

simpleType: serviceSubCodeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	N = None or Not Bulk
enumeration	E = Electronic
enumeration	R = Retail or Registered
enumeration	T = Requested at Time of Mailing
enumeration	B = Bulk

simpleType: standardFlatTypeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Catalog
enumeration	NotCatalog

simpleType: stateCode

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	AA
enumeration	AE
enumeration	AP
enumeration	AL
enumeration	AK
enumeration	AS
enumeration	AZ
enumeration	AR
enumeration	CA
enumeration	CO
enumeration	CT
enumeration	DE
enumeration	DC
enumeration	FM
enumeration	FL
enumeration	GA
enumeration	GU
enumeration	HI
enumeration	IL
enumeration	IN
enumeration	IA
enumeration	ID
enumeration	KS
enumeration	KY
enumeration	LA
enumeration	MH
enumeration	MD
enumeration	MA
enumeration	MO
enumeration	MI
enumeration	MN
enumeration	MS
enumeration	MT
enumeration	NE
enumeration	NH
enumeration	NV
enumeration	NJ
enumeration	NM
enumeration	NY
enumeration	NC

Tag	Mail.XML 11.0D
enumeration	ND
enumeration	MP
enumeration	OH
enumeration	OR
enumeration	OK
enumeration	PW
enumeration	PA
enumeration	PR
enumeration	RI
enumeration	SC
enumeration	SD
enumeration	TN
enumeration	TX
enumeration	UT
enumeration	VT
enumeration	VI
enumeration	VA
enumeration	WA
enumeration	WV
enumeration	WI
enumeration	WY

simpleType: statusType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Accepted
enumeration	Amended
enumeration	Cancelled
enumeration	Counterproposal
enumeration	New
enumeration	NoAction
enumeration	Original
enumeration	PartiallyAccepted
enumeration	Pending
enumeration	Rejected
enumeration	Replaced

simpleType: stopDesignatorType

Tag	Mail.XML 11.0D
Base	mailxml_base:s02

simpleType: surchargeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	2 = Oversized Surcharge
enumeration	A = First-Class Mail Parcel Surcharge
enumeration	D = Dimensional Rate

Tag	Mail.XML 11.0D
enumeration	N = Not Oversized
enumeration	P = Balloon Surcharge
enumeration	E = Permit Reply Mail
enumeration	R = First-Class Mail Non-machinable Letters (0.0625 lbs or less)
enumeration	L = Detached Address Labels Used

simpleType: totalWeightType

Tag	Mail.XML 11.0D
Base	mailxml_base:s05

simpleType: trailerLengthType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	20ft
enumeration	40ft
enumeration	45ft
enumeration	48ft
enumeration	53ft

simpleType: transportationMethodType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	USPS
enumeration	Mailer

simpleType: transportationModeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	0 = Reject Due to Error
enumeration	1 = Local and Working
enumeration	2 = Bypassed or Unassigned Surface or Air Mail
enumeration	3 = Assigned Surface (S-AMS Surface)
enumeration	4 = Assigned Air
enumeration	9 = Reject Due to Purge

simpleType: transShipBoLNumberType

Tag	Mail.XML 11.0D
Base	mailxml_base:s05

simpleType: trayType

Tag	Mail.XML 11.0D
Base	xs:string

Tag	Mail.XML 11.0D
enumeration	MixedLengthTrays
enumeration	1ft Tray
enumeration	2ft Tray
enumeration	EMM Tray
enumeration	Flat Tub

simpleType: trayTypeIndicatorType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	T = 2 Foot Tray
enumeration	M = MLOCR
enumeration	N = MM Tray
enumeration	E = EMM Tray
enumeration	O = 1 Foot Tray
enumeration	U = Unit Load Device
Refer to Mail.dat guide MD_11_2_FS_v11 2 0 41 page 53 for a complet list of container types from the Container summary Record – .csm	

simpleType: truckDispatchNumberType

Tag	Mail.XML 11.0D
Base	mailxml_base:s10

simpleType: unitLoadType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	CrossDockPallets
enumeration	WorkingPallets
enumeration	Trailers

simpleType: uOMType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	AirDryMetricTonne
enumeration	AirDryPercent
enumeration	AirDryShortTon
enumeration	Bale
enumeration	BoardFoot
enumeration	BookUnit
enumeration	Box
enumeration	Bundle
enumeration	C-Size
enumeration	Centimeter
enumeration	Cord
enumeration	CubicFoot
enumeration	CubicMeter
enumeration	Cubit

Tag	Mail.XML 11.0D
enumeration	Day
enumeration	DotsPerInch
enumeration	Foot
enumeration	Gallon
enumeration	Gram
enumeration	GramsPerSquareMeter
enumeration	Hour
enumeration	HundredBoardFeet
enumeration	HundredLinealFeet
enumeration	HundredPound
enumeration	HundredSquareFeet
enumeration	Inch
enumeration	Kilogram
enumeration	KilogramsPerDay
enumeration	KilogramsPerWeek
enumeration	KnownBreaks
enumeration	Layer
enumeration	Leaves
enumeration	LinearFoot
enumeration	LinesPerInch
enumeration	Load
enumeration	MagazineUnit
enumeration	Megabyte
enumeration	Meter
enumeration	MetricTon
enumeration	MetricTonsPerDay
enumeration	MetricTonsPerWeek
enumeration	Micron
enumeration	Millimeter
enumeration	Minute
enumeration	Month
enumeration	None
enumeration	Package
enumeration	Page
enumeration	PagesPerInch
enumeration	PalletUnit
enumeration	Percentage
enumeration	PerThousand
enumeration	Picas
enumeration	Piece
enumeration	PixelsPerInch
enumeration	Pound
enumeration	PoundsPerDay
enumeration	PoundsPerHour
enumeration	PoundsPerWeek
enumeration	PulpUnit
enumeration	Ream
enumeration	Reel
enumeration	Set
enumeration	Sheet

Tag	Mail.XML 11.0D
enumeration	ShortTon
enumeration	ShortTonsPerDay
enumeration	ShortTonsPerHour
enumeration	ShortTonsPerWeek
enumeration	Signature
enumeration	Skid
enumeration	SquareInch
enumeration	SquareFeet
enumeration	SquareMeter
enumeration	ThousandBoardFeet
enumeration	ThousandLinealFeet
enumeration	ThousandPieces
enumeration	ThousandSquareCentimeters
enumeration	ThousandSquareFeet
enumeration	ThousandSquareInch
enumeration	Ton
enumeration	TonsPerHour
enumeration	Unit
enumeration	UnknownBreaks
enumeration	Week
enumeration	Yard

simpleType: uniqueContainerIDType

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	12
minLength	1
whiteSpace	Preserve

simpleType: userInformationLine1Type

Tag	Mail.XML 11.0D
Base	mailxml_base:s40

simpleType: userInformationLine2Type

Tag	Mail.XML 11.0D
Base	mailxml_base:s40

simpleType: userLicenseCodeType

Tag	Mail.XML 11.0D
Base	xs:string
maxLength	4
minLength	1
whiteSpace	Preserve

simpleType: uspsPickupType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Y = Yes
enumeration	N = No

simpleType: verificationLocationType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	BMEU
enumeration	DMU

simpleType: verificationReportType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	CSAVerification
enumeration	ContainerBarcodeVerification
enumeration	HandlingUnitBarcodeVerification
enumeration	PieceBarcodeVerification
enumeration	ByForConflictVerification
enumeration	ServiceTypeVerification
enumeration	AppointmentVerification
enumeration	MailerIDVerification
Enumeration	ServicePerformanceVerification
enumeration	DeliveryPointVerification
enumeration	EntryFacilityVerification
enumeration	DefaultTrayBarcodeVerification
enumeration	IMDVerification
enumeration	All
enumeration	NoErrors

simpleType: wastedPieceIndicatorType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	W = Wasted
enumeration	S = Shortage
enumeration	X = Wasted piece and postage adjustment should NOT be triggered
enumeration	T = Shortage pieces not produced and postage adjustment should NOT be triggered

simpleType: weekFrequencyType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Weekly

Tag	Mail.XML 11.0D
enumeration	Bi-Weekly
enumeration	First
enumeration	Second
enumeration	Third
enumeration	Fourth
enumeration	Fifth
enumeration	Sixth

simpleType: weightBreakType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = For standard Mail Non-Letters, Piece Weight is over 0.2063 lbs. For First-Class Mail Permit Reply, piece weight is over 0.0625 lbs.
enumeration	B = For standard mail, piece weight is 0.2063 lbs or less For First-Class Mail Permit Reply, piece weight is 0.0625 lbs or less.
enumeration	H = For standard Mail Letters, piece weight is over 0.2063 lbs
enumeration	N = None

simpleType: yesNo

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Yes
enumeration	No

simpleType: zipCodeThreeOrFiveDigit

Tag	Mail.XML 11.0D
Base	xs:string
pattern	"[0-9]{3}|[0-9]{5}" - Space removed

simpleType: zoneICType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	N = None
enumeration	M = [Future Placeholder]
enumeration	LC = Local
enumeration	1 = Zone 1 & 2
enumeration	3 = Zone 3
enumeration	4 = Zone 4
enumeration	5 = Zone 5
enumeration	6 = Zone 6
enumeration	7 = Zone 7

Tag	Mail.XML 11.0D
enumeration	8 = Zone 8
enumeration	ICD
enumeration	IC
enumeration	DDU
enumeration	SCF
enumeration	ADC

simpleType: onePassOrTwoPassType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	One
enumeration	Two

simpleType: uspsPackageUniqueCharacteristicType

Tag	Mail.XML 11.0D
Base	Xs:string

simpleType: StandardFlatTypeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Catalog
enumeration	NonCatalog

simpleType: MarketingOrFulfillmentType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	Marketing
enumeration	Fulfillment

simpleType: MLOCRRateType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	R
enumeration	N

simpleType: DMLTypeType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A
enumeration	B

simpleType: retrieveDataBy

Tag	Mail.XML 11.0D

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	FS = Full Service
enumeration	NFS = Non-Full Service
enumeration	Both

simpleType: MID6

Tag	Mail.XML 11.0D
Base	xs:mailerID6Type

simpleType: MID9

Tag	Mail.XML 11.0D
Base	xs:mailerID6Type

simpleType: MPSQueryType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	A = All Detail
enumeration	S = Sample Detail
enumeration	SC = Scan Count

simpleType: MailPieceScanAvailabilityDate

Tag	Mail.XML 11.0D
Base	xs:date

simpleType: MailPieceScanRecordCount

Tag	Mail.XML 11.0D
Base	xs:nonNegativeInteger

simpleType: ScanState

Tag	Mail.XML 11.0D

Tag	Mail.XML 11.0D
Base	xs: MPSStateType

simpleType: LowerDateRange

Tag	Mail.XML 11.0D
Base	xs:date

simpleType: UpperDateRange

Tag	Mail.XML 11.0D
Base	xs:date

simpleType: MPSStateType

Tag	Mail.XML 11.0D
Base	xs:string
enumeration	1 = Electronic Shipping info Received
enumeration	2 = Entered at USPS
enumeration	3 = Enroute Arrive
enumeration	4 = Enroute
enumeration	5 = Enroute Departure

Appendix C - WSDLs and XSDs

The WSDL specification are posted on RIBBS® and can be downloaded from the following location: http://ribbs.usps.gov/intelligentmail_guides/documents/tech_guides/xmlspec/wsdls/wsdls.htm

The above RIBBS® URL provides WSDLs for both TEM and production environments. User should always use this link to access the correct and updated version of WSDLs in future.

PULL WSDL LINKS FOR TEM ENVIRONMENT
To consume the WSDL services, the customer must use one of the following URLs in the TEM environments:

STRING-TYPE WSDL URL Link(s)

The USPS is no longer supporting the string type as of the Mail.XML 9.0 implementation. Currently, the USPS only accepts Message-Types WSDLs.
MESSAGE-TYPE WSDL URL Link(s)

The following link supports Mail.XML Modular Specification (9.0 and above):

On the Modular XSD, the USPS supports the following WSDL points based on the Messages Group (Mailing, Supply Chain, Transportation, Data Distribution, and Identification). The WSDL endpoint for the message group will not change, will support multiple versions of Modular spec.

Mailing:
https://mailxmltem.uspspostalone.com/Mailing/WebServices/wsdl/Mailing.wsdl

Supply Chain:
https://mailxmltem.uspspostalone.com/SupplyChain/WebServices/wsdl/SupplyChain.wsdl

Transportation:
https://mailxmltem.uspspostalone.com/Transportation/WebServices/wsdl/Transportation.wsdl

Data Distribution:
https://mailxmltem.uspspostalone.com/DataDistribution/WebServices/wsdl/DataDistribution.wsdl

Identification:
https://mailxmltem.uspspostalone.com/Identification/WebServices/wsdl/Identification.wsdl

PULL WSDL LINKS FOR PRODUCTION ENVIRONMENT

To consume the WSDL services, the customer must use one of the following URLs in the Production environments:
STRING-TYPE WSDL URL Link(s)

The USPS is no longer supporting the string type as of the Mail.XML 9.0 implementation. Currently, the USPS only accepts Message-Types WSDLs.

MESSAGE-TYPE WSDL URL Link(s)

The following link supports Mail.XML Modular Specification (9.0 and above):

On the Modular XSD, we support following WSDL points based on the Messages Group (Mailing, Supply Chain, Transportation, Data Distribution, and Identification). The WSDL endpoint for the message group will not change, will support multiple versions of Modular spec.

Mailing:
https://p1webservices.uspspostalone.com/Mailing/WebServices/wsdl/Mailing.wsdl

Supply Chain:
https://p1webservices.uspspostalone.com/SupplyChain/WebServices/wsdl/SupplyChain.wsdl

Transportation:
https://p1webservices.uspspostalone.com/Transportation/WebServices/wsdl/Transportation.wsdl

Data Distribution:
https://p1webservices.uspspostalone.com/DataDistribution/WebServices/wsdl/DataDistribution.wsdl

Identification:
https://p1webservices.uspspostalone.com/Identification/WebServices/wsdl/Identification.wsdl

PUSH WSDLs INFORMATION

Customers interested in using the Push Subscription model to receive the data feedback automatically at a specified time to their servers must understand the following WSDL name change information for both string-type and message-type WSDLs:

STRING-TYPE PUSH WSDL

The USPS is no longer supporting the string type as of the Mail.XML 9.0 implementation. Currently, the USPS only accepts Message-Types WSDLs.

MESSAGE-TYPE PUSH WSDL

The customer need to provide URL for their Web Server and uses the following WSDL that only supports Mail.XML 11.0D.

For Mail.XML 11.0D – following WSDL name shall be used:
WSDL NAME: DataDistributionPushMailXML110D.wsdl

PUSH MESSAGES XSD NAME CHANGES INFORMATION

The USPS is implementing a new XSD name to support Mail.XML 11.0D Push subscription messages.

INDEX