

PostalOne! Mail.XML Technical Guide

Version 6.8.3

USPS Program Manager: Carla Siniscalchi (202) 268-4059

Business System Solutions Program Manager: Michael Haskin (314) 604-1452

Assurety Consulting Inc. Point of Contact: Shariq Mirza (703) 292-4106

Accenture Point of Contact: Eben Joel (703) 947-1911

***PostalOne!* IT Point of Contact:** Ibrahim Hamid (703) 248-3507

***PostalOne!* IT Point of Contact:** Jeffrey Avellanet (703) 248-3525

Table of Contents

Change History	4
1.0 POSTALONE! MAIL.XML TECHNICAL GUIDE	6
1.1 WHAT IS MAIL.XML?	6
<i>Mail.XML 6.0</i>	6
1.2 PURPOSE	7
2.0 POSTALONE! – FAST MAIL.XML WEB SERVICES	8
2.1 OVERVIEW	8
2.2 ACCESS TO <i>POSTALONE!</i> - FAST WEB SERVICES	8
2.3 <i>POSTALONE!</i> - FAST MAIL.XML WEB SERVICES SUPPORTED MESSAGES	8
<i>Mail.XML 4.0 Specifications</i>	9
<i>Mail.XML 6.0 Specifications (Full-Service Release 1 FAST Messages)</i>	11
2.4 ADDITIONAL RULES/CONSTRAINTS	15
<i>Detail Info Block</i>	25
2.5 MULTISTOP SUPPORT	25
<i>FAST Online Application</i>	25
<i>Web Services</i>	26
2.6 ERROR CODES/RETURN VALUES	26
2.7 CIN WARNING CODES AND DESCRIPTIONS	32
2.8 FAULT CODES	36
2.9 WEB SERVICE ADDITIONAL CONSTRAINTS	36
2.10 MESSAGE RESPONSE RETRIEVAL REQUEST PROCESSING	37
2.11 <i>POSTALONE!</i> WSDL INFORMATION	38
2.12 SCHEDULERCONTENT ID	38
2.13 CLOSEOUT MESSAGES	38
2.14 APPOINTMENT SHELL CANCELLATION MESSAGES	39
2.15 <i>POSTALONE!</i> - FAST WEB SERVICES AUTHENTICATION:	39
3.0 POSTALONE! – EDOC MAIL.XML MESSAGES	40
3.1 OVERVIEW	40
3.2 EDOC AND MISCELLANEOUS MESSAGE	40
3.3 MAIL.XML SPECIFIC MESSAGE DETAILS	41
<i>Mailing Group Query Request and Response</i>	41
<i>PaymentMessageQueryRequest and Response</i>	41
<i>PaymentMessageQueryResponse</i>	41
<i>Reconciliation Report Query Request and Response</i>	41
<i>Container Status Messages</i>	42
3.4 MESSAGE RESPONSE RETRIEVAL REQUEST PROCESSING	42
<i>Fault and Fault Recovery Messages</i>	42
3.5 ADDITIONAL RULES/CONSTRAINTS	43
<i>Container Status Messages</i>	43
3.6 ERROR CODES/RETURN VALUES	43
3.7 ADDITIONAL CONSTRAINTS	44
3.8 WSDL INFORMATION	44
4.0 FULL-SERVICE DATA DISTRIBUTION – RELEASE 1	45
4.1 PURPOSE	45
4.2 WHICH METHODS DOES THE <i>POSTALONE!</i> SYSTEM SUPPORT FOR FULL-SERVICE DATA FEEDBACK?	45
4.3 WHAT DO YOU NEED TO RECEIVE YOUR DATA THROUGH THE DATA EXCHANGE METHODS IDENTIFIED?	46
<i>Step 1</i>	46
<i>Step 2</i>	46
4.4 HOW DO YOU IMPLEMENT/USE THE MAIL.XML AVAILABLE FOR FULL-SERVICE DATA DISTRIBUTION?	46

	<i>Why Mail.XML-based Data Distribution?</i>	47
4.5	MESSAGES SUPPORTED IN RELEASE 1	47
4.6	PUSH MESSAGES	47
4.7	PULL MESSAGES	48
4.8	MESSAGES	48
	<i>Push Messages Supported in Release 1 of Full-Service Implementation:</i>	48
	<i>Pull Messages Supported in Release 1 of Full-Service Implementation:</i>	49
	<i>Cast Of Characters Messages for Release 1 of Full-Service Implementation:</i>	49
4.9	MESSAGE RESPONSE RETRIEVAL REQUEST PROCESSING.....	50
	<i>Fault and Fault Recovery Messages</i>	50
4.10	ADDITIONAL BUSINESS RULES.....	50
	<i>Delivery and Notification Messages:</i>	50
	<i>Request and Response Messages</i>	56
	<i>Cast Of Characters Messages for Release 1 of Full-Service Implementation:</i>	60
4.11	PULL REQUEST/RESPONSE-HANDLING LARGE TRANSACTIONS	64
4.12	ERROR CODES/RETURN VALUES	64
APPENDIX A - WSDLs AND XSDS.....		66
	<i>PUSH User Profile</i>	93
	<i>User Authentication Info XSD</i>	96
	<i>WSDL - Required for the PUSH Model to Receive Data from the PostalOne! System</i>	97
	<i>User Authentication XSD -- Required to allow the PostalOne! System to Invoke Customer's Web Services</i>	102
APPENDIX B – MESSAGE CHOREOGRAPHY FOR DATA DISTRIBUTION.....		103
APPENDIX C - EDOCUMENTATION FORM VALUES		105
	EDOC FORM VALUES.....	105
	<i>Container Status Type</i>	105
	<i>Entry Facility Type</i>	105
	<i>Form Type</i>	105
	<i>Full-Service Indicator</i>	105
	<i>Rate Category</i>	106

Change History

Date	Description	Version
2/24/2009	Updated guide Per changes from IDEAlliance for 6.0B	6.8.0
2/24/2009	Updated the Submitting Party in Section 3.2 to read "Optional – Either CRID or MID is required for Data Distribution or eDOC messages"	6.8.0
2/24/2009	Removed reference to Cast of Characters in Section 3.5	6.8.0
2/24/2009	Updated the Prerequisite in Section 4.4. to read "must establish push subscription profiles in the <i>PostalOne!</i> system"	6.8.0
3/16/2009	Made changes to the WSDLs for 6.0C changes in Mail.XMI made by IDEAlliance which fixed the IMB pattern error,	6.8.1
3/16/2009	Updated the Mail.XML version to Mail.XML 6.0 Errata – C version : page 5 Added additional error messages to the data distribution error codes : page 59	6.8.1
3/16/2009	Added detail descriptions for FullServiceAddressCorrection block for the delivery/notification message business rules. These updates include the list of detail information for elements Move Type, COA Name, Class Notification Type, Deliverability Code Type and Address Type.	6.8.12
3/16/2009	Added detail descriptions for FullServiceNixieDetail blocks for the delivery/notification message business rules. These updates include the list of detail information for elements Action Code, Class Notification Type and Reason Code	6.8.12
3/16/2009	Added detail descriptions for CastofCharactersCancelRequest for the delivery/notification message business rules. These updates include fields that can be used to cancel all or some of the original Cast of Characters records. These fields include OriginalSenderMsgId, SubmittingParty, MailPieceGroup, Participant, and Role element.	6.8.12
3/16/2009	Added detail descriptions for CastofCharactersCancelResponse & CastOfCharactersUpdateResponse for the delivery/notification message business rules. These updates include information that will be provided during the Accept or Reject of a Cancel/Update Request Message	6.8.12
3/16/2009	Added detail descriptions for	6.8.12

	CastofCharactersUpdateRequest for the delivery/notification message business rules. These updates include fields that should be used to send the Update Request message. These fields include OriginalSenderMsgID, Submitting Party, Submitting Software, MailPieceGroup, Participant, Role, and Effective Date information.	
3/16/2009	Added new error codes/return values for Cast of Characters Response messages. These new error codes are 6000, 6014, 6015, 6016, 6017.	6.8.12
3/16/2009	Clarified that job information elements Customer Group ID and Mailing Group ID pertains to both Mail.XML & Postal Wizard for Start-the-Clock and Address Correction messages.	6.8.12
3/30/2009	Made changes to the WSDLs for 6.0D changes by IDEAlliance, which fixed the IMB pattern error, increased the size of MailXMLappts to 10 bytes from 9 bytes and changed the Last Name field length to 50 bytes to allow sending of Business name in the Last Name field as a work around. Also, added more detail on ACS referewncereference codes to help new developers help with the process.	6.8.2
3/30/2009	Updated section 2.10 as follows: The PostalOne! system will generate the response to a MessageResponseRetrievalRequest (MRRR) for Mail.XML 6.0 requests and ApptResponseRetrievalRequest (ARRR) for Mail.XML 4.0 requests	6.8.2
4/1/2009	Added change history log	6.8.3

1.0 **PostalOne! Mail.XML Technical Guide**

1.1 **What is Mail.XML?**

The Mail.XML™ is an XML-based IDEAlliance® (www.mailxml.org) specification for web services supporting the (SOA) Services Oriented Architecture that is conversational in nature and is platform independent. Web Services uses simple HTTP(s) protocol to communicate data over the internet, hence bypassing technology specific restrictions and avoiding network related security rules. Sound implementations of web services are safe, secure, platform independent, support near real time communication, and increase efficiency of conducting business in most business environments.

More importantly, the Mail.XML allows business function-specific communication and faster change management processes with no impact to other verticals within the mailing industry, which has been a core improvement area for the Mail.dat® specification. Mail.XML is a complement to Mail.dat, where people using Mail.dat can continue to use it as a database but will use the Mail.XML for communication, automation of business to business processing and getting answers in near real time.

The Mail.XML 6.0 functionality, when implemented in May 2009, will have three major business functions: eDoc messages, data distribution messages, and FAST scheduling messages. These Mail.XML-based business functions will surpass any functionality available today through any data interface with the USPS®. The IDEAlliance Mail.XML version 6.0 specification is composed of families of related messages that provide all the capability that Mail.dat specification provided, as well as business functions support that Mail.dat did not provide. Business functions supported include appointment scheduling management, Cast of Characters communication, reception of full-service feedback data from the USPS, bi-directional data communication to conduct business with the USPS, and the ability to request Customer/Supplier Agreement (CSA), Customer Registration ID (CRID) and Mailer ID (MID) data. The specification also provides the capability for the industry to conduct many business functions within the industry through near real time communication mechanisms specifically to support joint scheduling and transportation management communications.

Mail.XML 6.0

Mail.XML version 6.0 enhances efficiency and ease of transmission for the small and large data volume communications. An attribute has been added to all large data messages that allows all large data volume messages to be broken up in as many unique child transactions as the USPS and the industry believes will be easily and efficiently 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. The Mail.XML allows for verification and comprehensive validation of the data, right away with a “feet ahead” concept, in near real time.

When the USPS receives data from you, USPS can respond in near real time to let you know the status of the transaction and status of the data received from you. 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. The bi-directional communication allows for automatic conflict resolution triggering that can result in immediate action by the USPS or you.

The IDEAlliance Mail.XML version 6.0 specification, currently scheduled for implementation by the USPS in multiple phases, will support full-service electronic documentation functionality including single piece information, range piece information, nesting of pieces into handling units and handling units into containers, postage payments, qualification reports, and electronic documentation management through transaction updates. The Mail.XML version 6.0 is used for providing data feedback to you for Full-service, such as StartTheClock, Container Scans, Address Change Service

(ACS) Change of Address (COA), and ACS Nixie data, as well as Verification errors such as issues with e8125 or e8017 or By/For information mismatches.

The previous versions of the Mail.XML already provide the support for the appointment scheduling business processes with the *PostalOne!*® FAST systems. The USPS currently supports Mail.XML 2.0 and Mail.XML 4.0 specifications in production and in May it will support the appointment management processes with Mail.XML 6.0 and Mail.XML 4.0.

This document focuses solely on the *PostalOne!* system's May 2009 implementation and further updates to this guide will be made for every phase of the Mail.XML implementation by the USPS.

The *PostalOne!* system plans to implement the Mail.XML 6.0 Errata-D version in May 2009.

Mail.XML 6.0 errata versions are scheduled for publication once a month, so check for Mail.XML errata versions on the www.mailxml.org site periodically and look for DMM notifications as well as the External Timeline document for IM implementation shared with the MTAC 122 work group to learn about any changes in the *PostalOne! Mail.XML Technical Guide*.

1.2 Purpose

The purpose of this Technical Guide is to supplement the IDEAlliance Mail.XML (previously known as IDEAlliance Mail.dat Transportation Management (TM) Specification) and to provide Mail.XML users with specific information regarding the USPS implementation. USPS will support Mail.XML: 6.0 as part of the Full-Service program in a phased approach with the first phase scheduled for the May 2009 release. 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 will be updated as IDEAlliance introduces or upgrades Mail.XML Specification. The USPS will make the return codes, error codes and their descriptions, as well the WSDLs for future phases of Mail.XML implementation available to the industry on the dates documented in the *External TimeLine for IM implementation* document made available to the MTAC 122 work group. So, check for an update to this guide at that time.

This guide is divided into four sections;

Section 2.0 provides technical information with regard to the Mail.XML version 6.0 messages that support the *PostalOne!* - FAST Web Services interface and can largely be grouped as scheduling-related messages. The *PostalOne!* system and FAST will continue to support two versions of the Mail.XML specifications; namely Mail.XML 4.0 and Mail.XML version 6.0 with the implementation of the Full-Service program in May.

Section 3.0 discusses Phase I of Mail.XML eDoc messages that are implemented in May, which is a list of eight messages. This section only discusses the May 2009 implementation.

Section 4.0 discusses the data distribution and data feedback push and pull models to receive StartTheClock, Container Visibility, ACS COA, and ACS Nixie Detail data.

Later sections provide WSDLs and XSDs that are specific to the USPS.

This guide will be updated as additional information is communicated by IDEAlliance or by the USPS IT with the release of the errata versions of the Mail.XML 6.0 specification.

2.0 *PostalOne!* – FAST Mail.XML Web Services

2.1 Overview

The USPS requires appointment scheduling for drop shipped mail at USPS facilities for better resource planning and service performance. The USPS FAST (Facilities Access and Shipment Tracking) system is the business process engine that provides mailers the capabilities to schedule appointments.

2.2 Access to *PostalOne!* - FAST Web Services

There are two methods to create, update, and cancel appointments and receive Closeout data within the FAST system. For both processes, a mail owner or mailing agent must request a user account via the *PostalOne!* system, which performs authentication and authorization services for FAST and *PostalOne!* systems. The two methods constitute:

1. Going online to <http://fast.usps.com>, logging in with your username and password and choosing FAST functional GUIs to request an appointment at a facility at a certain date and time. This process is manual process requiring manual data entry. This online FAST process does not require the customer to provide content (container) detail information, such as IM Container barcode, weight, pieces, and many other characteristics due to the manual nature of the process.
2. Utilizing the IDEAlliance Mail.XML (formerly “TM”) Specification and *PostalOne!*/FAST Web Services, where the *PostalOne!* system works as the data exchange gateway for the appointment transactions and passes the appointment and container detail information, including IM Container barcode, class, shape, piece count, weight, zone skipping, mail owner information and many other content characteristics to the FAST system and all messages to and from the customer pass through the *PostalOne!* system. The *PostalOne!* system can also be leveraged in this usage scenario to pull existing content (container detail) information from customer’s or customer’s business partner’s previously provided Mail.dat to the *PostalOne!* system. The customer can either send all the content details directly to FAST through the *PostalOne!*/FAST Web Services mechanism or can request the *PostalOne!* system to query and provide all or part of the content details to FAST through the *PostalOne!* FAST Web Services mechanism.

The Full-Service program requires customers to use the IDEAlliance Mail.XML messages and *PostalOne!*/FAST Web Services to create and/or update appointments within the FAST system for one-time appointments. One-time appointments created through the online FAST application **must be updated electronically** through Mail.XML with the Intelligent Mail Container Barcode (IMCB) to receive Full-Service discounts. Mail.XML 6.0 can be used to create or update one time, recurring, drop shipment and origin entry appointments.

2.3 *PostalOne!* - FAST Mail.XML Web Services Supported Messages

The *PostalOne!* - FAST Mail.XML Web Services interface will continue to support two versions of the Mail.XML Specifications at a given time. *PostalOne!* - FAST Mail.XML Web Services will support Mail.XML v4.0 specifications and Mail.XML version 6.0 specifications with the release of Full-Service.

The sections below provide information about the subset of Mail.XML messages that will be supported by both the *PostalOne!* system and FAST in Mail.XML 4.0 and subsequently by Mail.XML 6.0.

Mail.XML 4.0 Specifications

Mail.XML 4.0 Messages Supported by the *PostalOne!* system and FAST – Already in Production as of November 16, 2008

The *PostalOne!* - FAST Mail.XML Web Services will support the following groups of messages in Mail.XML 4.0:

Delivery Appointment Messages

This family of Mail.XML transportation related messages supports scenarios where a Shipper requests a Delivery Appointment from the Consignee. The Consignee may respond a number of ways, all of which are supported by messages in this message family. Delivery appointment messages will:

- Provide the ability to create, update, and cancel drop ship appointments
- Provide the ability to create, update, and cancel First-Class Mail® DMU customer-transported appointments
- Provide the ability to query the availability of slots for drop ship appointments
- Provide the ability to query the availability of slots (based on hours of operation) for First-Class DMU/BMEU customer-transported appointments
- Provide the ability to distinguish eVS parcels and submit the associated eFile Number for which they were manifested
- Provide the ability to support new container types and handling units
- Provide the ability to support sibling containers
- Provide the ability to associate stand-alone contents to a one-time appointment
- Grant the ability to provide the Intelligent Mail Container Barcodes (IMCB) associated with the appointment

Delivery Appointment Messages

DeliveryApptCreateRequest / DeliveryApptCreateResponse

DeliveryApptUpdateRequest / DeliveryApptUpdateResponse

DeliveryApptCancelRequest / DeliveryApptCancelResponse

DeliveryApptQueryRequest / DeliveryApptQueryResponse

Delivery Content Messages

This family of Mail.XML transportation related messages supports scenarios where the *PostalOne!* system sends content information to FAST without sending any appointment-level information. The Consignee may respond a number of ways, all of which are supported by messages in this message family. These sets of messages apply for the scheduling of drop ship appointments only.

Delivery Content messages provide the:

- Ability to create, update, and cancel stand-alone content
- Ability to create, update, and cancel contents for one-time and recurring appointments

- Ability to distinguish eVS parcels and submit the associated eFile Number for which they were manifested
- Support new container types and handling units
- Support sibling containers
- Ability for schedulers to query for content created by other mail owners/mail preparers
- Ability to provide the Intelligent Mail Container Barcodes (IMCB) associated to the appointment

Delivery Content Messages

DeliveryContentCreateRequest / DeliveryContentCreateResponse

DeliveryContentUpdateRequest / DeliveryContentUpdateResponse

DeliveryContentCancelRequest / DeliveryContentCancelResponse

DeliveryContentQueryRequest / DeliveryContentQueryResponse

Appointment Shell Messages

This family of Mail.XML transportation related messages enables mailers to create appointment shells (appointments with no associated content). Mail owners and mail preparers can use the Delivery Content Messages to create and associate content to appointment shells. This family of messages allows for joint scheduling between schedulers and mail owners and mail preparers. Appointment shell messages will provide the:

- Ability to create, update, and cancel recurring appointment shells
- Ability to create, update, and cancel one-time appointment shells
- Ability to update the logistics information of an appointment by utilizing the appointment shell update message

Appointment Shell Messages

DeliveryApptShellRequest / DeliveryApptShellResponse

DeliveryApptShellUpdateRequest / DeliveryApptShellUpdateResponse

DeliveryApptShellCancelRequest / DeliveryApptShellCancelResponse

Recurring Appointment Query Messages

This family of Mail.XML transportation related messages allows mailers to query for active and pending recurring appointments. Recurring Appointment query messages will provide the:

- Ability to query for active and pending drop ship recurring appointments
- Ability to query for active First-Class DMU/BMEU verified customer transported appointments

Recurring Appointment Query Messages

RecurringApptQueryRequest / RecurringApptQueryResponse

Appointment Closeout Messages

This series of messages is pushed from the Consignee to the Shipper when an appointment is closed. Closeout Messages will provide the:

- Ability to send closeout information for First-Class DMU/BMEU verified customer-transported appointments
- Ability to send closeout information for drop ship appointments
- Ability to communicate closeout irregularities
- Ability to communicate arrival date/time, unload start date/time, unload end date/time, and actual pallet counts
- Ability to indicate if IMCB Pallet Placard scan occurred
- Ability to communicate if a drop ship appointment will be excluded from rating

Appointment Closeout Messages

AllDeliveryAppointmentCloseoutRequest

ConsigneeGoodsReceipt

UnscheduledConsigneeGoodsReceipt

Mail.XML 6.0 Specifications (Full-Service Release 1 FAST Messages)

The following major enhancements will be incorporated in FAST with IDEAlliance Mail.XML 6.0:

- Provide the ability to create, update, and cancel Standard Mail®, Periodicals, and Package Services class-based appointments for customer-transported origin entry appointments
- Provide the capability to create, update, cancel recurring appointments for customer transported origin entry appointments (First-Class Mail, Standard Mail, Periodicals, and Package Services).
- Provide the capability to create, update, cancel recurring appointments for drop ship appointments
- Provide the capability to query recurring appointments for drop shipments and origin entry appointments for a scheduler or for the entire corporation
- Provide the capability to associate more than 1 sibling container with barcode to a parent container barcode.
- Leverage the Delivery Appointment Cancel Create message to move containers from one appointment to another when the original appointment was a “No Show”.
- Leverage the DeliveryContent messages to add, update, or cancel a container to an appointment.
- Leverage the Delivery Content Cancel message to release containers from a “No Show” appointment when the appointment status is “No Show” or the time of update is within 1 hour of appointment date/time.
- Provide closeout information for customer-transported origin entry and drop ship appointments.
- Leverage the Partner Appointment Query message to query appointment and content information on behalf of partner mailers for one-time appointments of origin entry and drop ship appointments
- Leverage the Customer / Supplier Agreements (CSAs) Query message to query CSA's
- Provide the capability to perform joint scheduling between two partners, one creating the appointment and the other associating content to it. This only applies to drop ship appointments scheduling only.

Messages Supported by the *PostalOne!* system and FAST in Mail.XML 6.0 – Full-Service Release 1

DeliveryApptCreateRequest	DeliveryApptCreateResponse
DeliveryApptUpdateRequest	DeliveryApptUpdateResponse
DeliveryApptCancelRequest	DeliveryApptCancelResponse
DeliveryApptQueryRequest	DeliveryApptQueryResponse
DeliveryContentCreateRequest	DeliveryContentCreateResponse
DeliveryContentUpdateRequest	DeliveryContentUpdateResponse
DeliveryContentCancelRequest	DeliveryContentCancelResponse
DeliveryContentQueryRequest	DeliveryContentQueryResponse
PartnerApptQueryRequest	PartnerApptQueryResponse
CustomerSupplierAgreementQueryRequest	CustomerSupplierAgreementQueryResponse
AllDeliveryAppointmentCloseoutRequest	AllAppointmentCloseoutResponse
DeliveryApptShellCreateRequest	DeliveryApptShellCreateResponse
DeliveryApptShellUpdateRequest	DeliveryApptShellUpdateResponse
DeliveryApptShellCancelRequest	DeliveryApptShellCancelResponse
ConsigneeGoodsReceipt	UnscheduledConsigneeGoodsReceipt
RecurringApptQueryRequest	RecurringApptQueryResponse
DeliveryAppointmentCancelCreateRequest	DeliveryAppointmentCancelCreateResponse

Delivery Appointment Messages

This family of Mail.XML transportation related messages supports scenarios where a Shipper requests a Delivery Appointment from the Consignee. The Consignee may respond a number of ways, all of which are supported by messages from this message family. Delivery appointment messages will:

- Provide the ability to create, update, and cancel customer transported origin entry appointments (First-Class Mail, Standard Mail, Periodicals, and Package Services)
- Provide the ability to create, update, and cancel customer transported origin entry recurring appointments (First-Class Mail, Standard Mail, Periodicals, and Package Services)
- Provide the ability to create, update, and cancel drop ship appointments
- Provide the ability to create, update, and cancel drop ship recurring appointments
- Provide the ability to query the availability of slots (based on hours of operation) for origin entry customer transported appointments
- Provide the ability to query for a list of available slots for drop ship appointments (based on appointment type, date range, and corporate rating of customer) for a facility
- Provide the ability to distinguish eVS parcels and submit the associated eFile Number for which they were manifested
- Provide the ability to support for new container types and handling units
- Provide the ability to support more than one sibling barcode
- Provide the ability to associate stand-alone contents to a one-time appointment
- Ability to provide the Intelligent Mail Container Barcodes(IMCB) associated to the appointment

- Provide the ability to cancel one or more appointments and create a new appointment using the delivery appointment cancel create message.
- Provide the ability to transfer container barcodes from one or more open or No Show appointments to a newly created appointment using the delivery appointment cancel create message.

Delivery Appointment Messages

DeliveryApptCreateRequest / DeliveryApptCreateResponse

DeliveryApptUpdateRequest / DeliveryApptUpdateResponse

DeliveryApptCancelRequest / DeliveryApptCancelResponse

DeliveryApptQueryRequest / DeliveryApptQueryResponse

DeliveryApptCancelCreateRequest / DeliveryApptCancelCreateResponse

Delivery Content Messages

This family of Mail.XML transportation related messages supports scenarios where the *PostalOne!* system sends content information to FAST without sending any appointment level information. The Consignee may respond a number of ways, all of which are supported by messages from this message family. These sets of messages apply for scheduling of drop ship appointments only.

Delivery Content messages provide the:

- Ability to create, update, and cancel stand-alone content that can be used by a business partner with a DeliveryAppointmentCreate or update Request
- Ability to create, update, and cancel contents for one-time and recurring appointments (drop ship only)
- Ability to distinguish eVS parcels and submit the associated eFile Number for which they were manifested
- Support for new container types and handling units
- Support for sibling containers
- Ability for schedulers to query for content created by other mail owners/mail preparers
- Ability to provide the Intelligent Mail Container Barcodes(IMCB) associated to the appointment
- Ability to add, update, or cancel a container to an appointment by a Scheduler

Delivery Content Messages

DeliveryContentCreateRequest / DeliveryContentCreateResponse

DeliveryContentUpdateRequest / DeliveryContentUpdateResponse

DeliveryContentCancelRequest / DeliveryContentCancelResponse

DeliveryContentQueryRequest / DeliveryContentQueryResponse

Appointment Shell Messages

This family of Mail.XML transportation related messages enables mailers to create appointment shells (appointments with no associated content). Mail owners and mail preparers can use the Delivery Content Messages to create and associate content to appointment shells. This family

of messages allows for joint scheduling between schedulers and mail owners and mail preparers for drop ship scenarios. Appointment Shell messages will provide the:

- Ability to create, update, and cancel recurring appointment shells
- Ability to create, update, and cancel one-time appointments shells
- Ability to update the logistics information of an appointment by utilizing the appointment shell update message
- Ability to leverage the Appointment Shell Update message to update logistics information (one-time and recurring) of an appointment when no content updates are necessary

Appointment Shell Messages

DeliveryApptShellCreateRequest / DeliveryApptShellCreateResponse
DeliveryApptShellUpdateRequest / DeliveryApptShellUpdateResponse
DeliveryApptShellCancelRequest / DeliveryApptShellCancelResponse

Recurring Appointment Query Messages

This family of Mail.XML transportation related messages allows mailers to query for active and pending recurring appointments. Recurring Appointment query messages will provide the:

- Ability to query for active and pending drop ship recurring appointments
- Ability to query for active origin entry DMU/BMEU verified customer transported appointments

Recurring Appointment Query Messages

RecurringApptQueryRequest / RecurringApptQueryResponse

Appointment Closeout Messages

This series of messages is pushed from the Consignee to the Shipper closeout an appointment. Closeout Messages will provide the:

- Ability to send closeout information for origin entry DMU/BMEU verified customer transported appointments
- Ability to send closeout information for drop ship appointments
- Ability to communicate closeout irregularities
- Ability to communicate arrival date/time, unload start date/time, unload end date/time, and actual pallet counts
- Ability to indicate if IMCB Pallet Placard scan occurred
- Ability to communicate if a drop ship appointment will be excluded from rating

Appointment Closeout Messages

AllDeliveryAppointmentCloseoutRequest
AllAppointmentCloseoutResponse
ConsigneeGoodsReceipt
UnscheduledConsigneeGoodsReceipt

Partner Appointment Query Messages

This family of Mail.XML transportation related messages allows mailers to query appointment and content information on behalf of partner mailers for drop ship one-time appointments:

- Partner Appointment Query Request
- Partner Appointment Query Response

Partner Appointment Query Messages

PartnerApptQueryRequest

PartnerApptQueryResponse

Customer Supplier Agreement Messages

This will allow mailers using Mail.XML to query customer supplier agreements via Web Services when a CRID or CSA ID is provided. For valid requests, FAST will provide the appropriate CSA data, else an error would be generated. This request could result in one or more CSA responses. These messages include:

- Customer Supplier Agreement Query Request
- Customer Supplier Agreement Query Response

CSA Messages

CustomerSupplierAgreementQueryRequest

CustomerSupplierAgreementQueryResponse

2.4 Additional Rules/Constraints

The following section outlines the specific rules and constraints for USPS implementation. For additional details regarding the calculations used for Appointment Rating, online supported functionality vs. Web Services supported functionality, etc., please refer to the *FAST User Guide*.

- A mailer must submit an appointment using the facility's local time (taking time zone and daylight savings into consideration) and FAST will respond using the same logic.
- The following outlines the list of possible values for Appointment Types, Mail Classes, Processing Categories, Presort Levels, and Rate Types for the Delivery Appointment Query Request, Delivery Appointment Create Request and Delivery Appointment Update Request. Also included in the Presort list is a breakdown of whether the content tied to that Presort will be treated as Working or Cross Dock for volume availability calculations.

Appointment Type (apptTypeType)		
Code	Description	Supported?
1	Pallet	Yes
2	DropAndPick	Yes
3	Speedline	Yes
4	Bedload	Yes

Mail Class (mailClassType)		
Code	Description	Supported?
1	First-Class Mail	Yes
2	Periodicals	Yes

3	Standard Mail	Yes
4	Packaged Services	Yes
6	Standard/Periodicals Co-Mailings	Yes*

* **NOTE:** Standard Mail/Periodicals Co-Mailings content will be processed by FAST as Standard Mail content. Currently, the *PostalOne!* system does not support Co-Mailing Class “6” due to postage implications. As a result, the *PostalOne!* system will not populate Mail Class “6” from Mail.dat files into the Mail.XML specification because files with that mail class fail validation.

Processing Category (processingCategoryType)		
Code	Description	Supported?
LT	Letter	Yes
FL	Flat	Yes
CD	Card	Yes*
PF	Parcel, First-Class Mail	Yes
MP	Machineable Parcel	Yes
IR	Irregular Parcel	Yes
OS	Outside Parcel	Yes
CM	Custom Mail	No
NA	Not Flat-Machinable < 6oz	Yes
NB	Not Flat-Machinable >= 6oz	Yes
NP	Non Machinable Parcels	Yes

NOTE: Processing Category of “CD” (“Card”) is translated to “LT” (“Letter”) for volume availability.

PreSort Types (containerLevelType)			
Code	Description	Supported?	Type
A	CR-Direct	Yes	Cross Dock
B	Mixed CR in 5 Digit	Yes	Cross Dock
C	Mixed CR in 3 Digit	Yes	Working
D	CR – 5D Scheme	Yes	Cross Dock
E	DPC/3D Scheme	No	N/A
F	RDC/Scheme	No	N/A
G	5 Digit (Auto/Presort)	Yes	Working
H	5 Digit (Merged)	Yes	Working
I	5 Digit (Presort Only)	Yes	Working
J	5 Digit (Barcode Only)	Yes	Working
K	Metro Scheme	Yes	Working
L	Mixed RDC	No	N/A
M	5D Scheme (Presort)	Yes	Cross Dock
N	5D Scheme (Auto/Presort)	Yes	Cross Dock
P	5D Scheme (Barcode)	Yes	Cross Dock
Q	5D Scheme (Merged)	Yes	Cross Dock
R	3 Digit (Auto/Presort)	Yes	Working
S	3 Digit (Barcode)	Yes	Working
T	3 Digit (Presort)	Yes	Working
U	3 Digit (CR/Auto/Presort)	Yes	Working

PreSort Types (containerLevelType)			
Code	Description	Supported?	Type
V	3 Digit Scheme	Yes	Working
W	Unique 3 Digit	Yes	Working
X	SCF	Yes	Working
Y	Protected SCF	Yes	Working
Z	ADC	Yes	Working
AA	AADC	Yes	Working
AB	Mixed ADC	Yes	Working
AC	Mixed AADC	Yes	Working
AD	ASF	Yes	Working
AE	BMC	Yes	Working
AF	Protected BMC	Yes	Working
AG	Mixed BMC	Yes	Working
AJ	Single Piece	No	Working
BA	Urban Direct	No	N/A
BB	Rural Direct	No	N/A
BC	Station	No	N/A
BD	City	No	N/A
BE	FSA	No	N/A
BF	DCF	No	N/A
BG	FCP	No	N/A
BH	Province	No	N/A
BI	Residual	No	N/A
BJ	Foreign	No	N/A
BK	Country	No	N/A
BL	Mixed Country	No	N/A
BM	M Bags	No	N/A

Rate Type (rateTypeType)		
Code	Description	Supported?
B	Bound Printed Matter	Yes
D	Parcel Select	Yes
F	Media	Yes
L	Library	Yes
P	Parcel Post	No**

** Parcel Post is not supported for drop ship appointments but is supported for Origin Entry appointments.

Rate Type is required for all Package Services content.

NOTE: The mailClassType, processingCategoryType, containerLevelType, and RateType fields are all contained in the USPS Summarization block. Each Delivery Appointment Query Request, Delivery Appointment Create Request, Delivery Appointment Update Request, Delivery Content Create Request and Delivery Content Update Request can contain more than one USPS Summary block. If a single USPS Summary block within these messages contains

an unsupported or invalid type for any of these fields, the entire message will be rejected. In addition, a Return Code/Description will be returned for each violation.

- Only the following fields will be populated with new data in the response messages. All other information will simply be populated based on what was originally sent in the request message.

Delivery Appointment Query Response

- All available appointment(s) are populated in a one to many Available Appt element in the ApptBlock block. (Mail.XML 4.0)
- All available appointment(s) are populated in a one to many AvailableHours element in the AvailableSlots block. (Mail.XML 6.0)
- ReturnInfo block containing Drop Entry Warnings if any have occurred
- If no Appointment matches are found, the ReturnInfo block containing one to many return code/description combinations is populated.

Delivery Appointment Create Acceptance Response

- The ScheduledAppt element in the ApptBlock block is populated.
- The ConsigneeApptID in the ResponseMsgHeaderInfo block is populated.
- The ConsigneeMultiStopID in the ResponseMsgHeaderInfo block is populated if the Appointment is tied to a Multistop.
- ReturnInfo block containing Drop Entry Warnings if any have occurred

Delivery Appointment Create Rejection Response

ReturnInfo block containing one to many return code/description combinations is populated.

Delivery Appointment Update Acceptance Response

- The ScheduledAppt element in the ApptBlock block is populated.
- ReturnInfo block containing CIN Warnings if any have occurred

Delivery Appointment Update Rejection Response

ReturnInfo block containing one to many return code/description combinations is populated.

Delivery Appointment Cancel Response

The CancelConfirmID element is populated if the Appointment was canceled successfully. Otherwise, a ReturnInfo block containing one to many return code/description combinations is populated.

Delivery Appointment Cancel Create Acceptance Response

- The ScheduledAppt element in the ApptBlock block is populated.
- The ConsigneeApptID in the ResponseMsgHeaderInfo block is populated.
- ReturnInfo block containing Drop Entry Warnings if any have occurred.
- Delivery Appointment Cancel Create Rejection Response
- ReturnInfo block containing one to many return code/description combinations is populated.

Delivery Content Accept Response

- The ConsigneeContentID in the ContentUSPSSummaryUpdate block is populated.
- ReturnInfo block containing Drop Entry Warnings if any have occurred.

Delivery Content Reject Response

The ReturnInfo block containing one to many return code/description combinations is populated.

Delivery Content Cancel Response

- A CancelConfirmID is populated if the content is cancelled.
- If the content cannot be cancelled, the ReturnInfo block containing one to many return code/description combinations is populated.

Delivery Appointment Shell Accept Response

- The PreferredAppt element in the ApptBlock block is populated with the actual appointment date/time of the Scheduled appointment.
- The ConsigneeApptID in the ResponseMsgHeaderInfo block is populated.
- 9-digit numeric ID will be returned for one-time appointment
- 5-digit numeric ID followed by the "R" character will be returned for recurring sequences
- The OptionalAppt field is not returned in the response message.

Delivery Appointment Shell Reject Response

ReturnInfo block containing one to many return code/description combinations is populated.

Delivery Appointment Shell Cancel Response

- A CancelConfirmID is populated if the appointment shell is cancelled.
- If the appointment cannot be cancelled, the ReturnInfo block containing one to many return code/description combinations is populated.

Delivery Content Query Response

- A ContentUSPSSummary block for each active content is populated.
- If no Appointment matches are found, the ReturnInfo block containing one to many return code/description combinations is populated.
- 5-digit Scheduler IDs will be returned in place of full Scheduler IDs for all mail preparers and mail owners. A full Scheduler ID will be returned if the Scheduler ID is the same as the requestor's Scheduler ID or the creator's Scheduler ID.

Recurring Appointment Query Response

- A RecurringApptBlk block for each active or pending recurring sequence is populated.
- A USPSSummary block for each active content is populated.
- If no Recurring Appointment matches are found, the ReturnInfo block containing one to many return code/description combinations is populated.

- 5-digit Scheduler IDs will be returned in place of full Scheduler IDs for all mail preparers and mail owners. A full Scheduler ID will be returned if the Scheduler ID is the same as the requestor's Scheduler ID or the creator's Scheduler ID

Partner Appointment Query Response

- Scenarios/Permissions
 - **Scheduler** - If the Submitting Party Scheduler ID corresponds to the Scheduler of the appointment, FAST will return all logistics and content information. A partial Scheduler ID (first 5-digits) will be returned for the Scheduler, Mail Owner(s) and Mail Preparers.
 - **Mail Preparer** - If the Submitting Party Scheduler ID corresponds to the Mail Preparer of the appointment or within the Mail Preparer's corporation, FAST will return all logistics information and content information for which they are the Mail Preparer. A partial Scheduler ID (first 5-digits) will be returned for the Mail Preparer, Mail Owner(s) and Scheduler.
 - **Mail Owner** - If the Submitting Party corresponds to the Mail Owner of the appointment or within the Mail Owner's corporation, FAST will return all logistics information and content information for which they are the Mail Owner. A partial Scheduler ID (first 5-digits) will be returned for the Mail Owner, Scheduler and Mail Preparer.
- An ApptBlk block for each appointment is populated.
- A ContentUSPSSummary block for each active content is populated.
- 5-digit Scheduler IDs will be returned in place of full Scheduler IDs for all mail preparers and mail owners.
- Customer / Supplier Agreement Query Response
 - A CSAInfo block for each pending, active or inactive Customer / Supplier Agreement is populated.
 - If no Customer / Supplier Agreement matches are found, the ReturnInfo block containing one to many return code/description combinations is populated.
 - If Customer Supplier Agreement ID (CSAID) is not provided, FAST will return all CSA's associated to the Owner CRID.
- The IDEAlliance specification allows for an unlimited number of USPS Summary blocks to be supplied in the Detail Information block. However, the maximum number of USPS Summary blocks allowed in any given message is 100.
- The IDEAlliance spec allows a Pallets block within a given USPSSummary block to contain both Presort counts and non Presort counts. For the USPS implementation, all non Presort counts must be added together and summarized into a single Pallets block. For example:

The IDEAlliance spec allows for the following:

Pallets Block 1: pallet count = 17

Presort 1: 5 digit, Count = 4

Presort 2: 3 digit, Count = 7

The USPS implementation requires the pallets to be broken into two Pallets blocks:

Pallets Block 1: Count = 11

Presort 1: 5 digit, Count = 4

Presort 2: 3 digit, Count = 7

Pallets Block 2: Count = 6; No presort records

NOTE: If the total presort counts do not equal the pallet block count for a pallet block, an error message will be returned.

- The IDEAlliance specification allows actual piece count to be provided for presort and non presort counts. FAST will only use the actual piece count for volume if it is provided for all presorts and non presorts for a given USPSSummary block.

FAST will use the Actual Piece count for the following USPSSummary block scenario:

Pallets Block 1: Count = 10

Presort 1: 5 digit, Count = 5, Actual Piece Count = 100

Presort 2: 3 digit, Count = 5, Actual Piece Count = 100

FAST will NOT use the Actual Piece count for the following USPSSummary block scenario:

Pallets Block 1: Count = 10

Presort 1: 5 digit, Count = 5, Actual Piece Count = 100

Presort 2: 3 digit, Count = 5, Actual Piece Count Not Provided

FAST will NOT use the Actual Piece count for the following USPSSummary block scenario:

Pallets Block 1: Count = 5, Actual Piece Count = 100

Pallets Block 2: Count = 5, Actual Piece Count Not Provided

- The IDEAlliance specification allows multiple UniqueContainerBarcode elements to be included in a single USPS Summary block. The USPS implementation will store all UniqueContainerBarcode elements and will validate for uniqueness. Uniqueness is validated based on a 45-day timeframe. UniqueContainerBarcodes associated to an appointment that was scheduled for 45 days in the past or for an appointment that has been cancelled or rejected are considered valid for re-use. If an appointment is no-showed, all associated UniqueContainerBarcodes are valid again after 12 days (288 hours) for drop ship appointments, and after 5 days (120 hours) for Origin Entry appointments.
- The IDEAlliance specification allows multiple SiblingBarcode elements to be included in a single USPS Summary block. The USPS implementation will store all SiblingBarcode elements and will validate for uniqueness. Uniqueness is validated based on a 45-day timeframe.
- The IDEAlliance specification allows for the SiblingBarcode and the UniqueContainerBarcode to be between 1 and 24 characters. FAST requires that the SiblingBarcode and the UniqueContainerBarcode be 21 characters and begin with "99M".
- The IDEAlliance specification allows multiple OptionalAppt elements to be included in a single Appointment block. The USPS implementation will only use the first OptionalAppt element and others will be ignored. In addition, the OptionalAppt element will only be supported for the Delivery Appointment Query Request message.
- The Facility Number should be populated with the facility Locale Key (characters 2-7 of the drop site key)
- The VerificationZip4 element in the Mail.dat Summarization block is not required in the IDEAlliance specification. However, it is a required element for the USPS implementation when using the Mail.dat Summarization block because the VerificationZip4 element must be provided in order to retrieve the actual Mail.dat data.
- Before any message is sent that requires the *PostalOne!* system to populate information from Mail.dat data, the Shipper must ensure the Mail.dat data that corresponds to the Appointment is up to date in the *PostalOne!* system prior to sending a Web Services request.

- The Perishable element is a required field in the USPSSummary block. It does not exist in the Mail.dat. The *PostalOne!* system will set the Perishable element value to 'No' when the *PostalOne!* system populates an USPSSummary block with Mail.dat data.
- For the Customer Registration Process, the user belonging to a Shipper must identify and verify his/her company's correct corporate headquarters (HQ) location ZIP (e.g., 22202-6761-000). The *PostalOne!* system will generate a 12-byte field for that 9 digit Scheduler Corporate ZIP+4, e.g., 222026761000. The last three digits of a Scheduler Corporate ID will always be 000 (three zeros). This 'SchedulerCorporateID' will be provided by the *PostalOne!* system to all shippers via e-mail and it will be sent to the *PostalOne!* system with every Web Services transaction by the shipper. Each User must also be assigned a 'SchedulerID' by the *PostalOne!* system, which will be used to uniquely identify the scheduler entity. The 'SchedulerID' will be generated in the background and shown to the user on the screen.

The Scheduler ID will also be e-mailed to the user/Scheduler/Shipper, so Shipper can always ensure the correct 'SchedulerID' is always part of every Web Services message that is sent to USPS for identification purposes. The Scheduler entity can be an internal business role within a Shipper's organization and will be tied to the Shipper's corporate headquarters for FAST to rate the Shipper's corporation. The format of the Scheduler ID will be location ZIP+4 format plus three bytes (201486791001). A unique Scheduler ID will be generated in the ZIP+4 format with appended sequence numbers (always three digits and these three digits will never be three zeros) and communicated to the user via e-mail each time for a given Scheduler entity.

For example, the *PostalOne!* system will use the following example format to identify Corp HQ and Scheduler Locations:

- 222021234000 - Corporate HQ (saved without the hyphen). Shipper **MUST not** use a hyphen (-) when sending the SchedulerCorpID to USPS). The first nine digits will be the Corporate HQ ZIP+4 and the last three digits will be sequence numbers and will always be three zeros (000).
- 222021234100 - SchedulerID or Scheduler Entity ID (business role) at a given Shipper location (saved without the hyphen). Shipper **MUST not** use a hyphen (-) when sending the SchedulerID to USPS. The last three digits will be sequence numbers and will never be three zeros.
- Do not send the same request repeatedly in succession if you did not receive an immediate response. Sending the message repeatedly with the same ShipperApptRequestID for a given SchedulerID will result in an error. If you received a Tracking ID from the *PostalOne!* system, use that Tracking ID to retrieve your message in the first response.
- The DeliveryContent messages can be used by Schedulers, Mail Owners and Mail Preparers to add/update/cancel drop ship content:
 - Mail.XML 6.0 DeliveryContent messages will be designated as drop ship if the DestinationDiscountIndicator is set to 'Yes', if DestinationDiscountIndicator is set to 'No', FAST would generate an error response.
 - Stand-alone content - FAST shall treat the content as stand-alone content if the ConsigneeApptID is not provided in the request message. Stand-alone contents can be leveraged only for drop ship scenarios.
 - One-time content for an existing appointment - FAST shall treat the content as one time content if the ConsigneeApptID provided in the request message is a 9-digit appointment ID.
 - Recurring content for an existing recurring sequence - FAST shall treat the content as recurring content if the ConsigneeApptID provided in the request message is a 5-digit recurring sequence ID followed by the character "R".

- Mail.XML 6.0 DeliveryContent messages can be used to add/update/remove a specific content for an appointment created by the Scheduler.
- The DeliveryContentUpdateRequest and DeliveryContentCancelRequest messages may only be used to update or cancel content that was created using the DeliveryContentCreateRequest message.
- Mail.XML 6.0 DeliveryContentCancelRequest messages may be used to cancel specific container barcodes scheduled for induction but not inducted with the appointment.
 - Multiple contents may be canceled by providing each content's ConsigneeContentID (and ConsigneeApptID for associated content) for Mail.XML 6.0 DeliveryContentCancel requests.
 - If a Mail.XML 6.0 DeliveryContentCancelRequest is sent and the appointment date/time is already past the 1 hour update threshold, FAST will release all active containers associated to the specified content(s). (Please note that contents themselves are not "released").
- The DeliveryContentQuery messages may be used to query content for drop ship appointments:
 - If ConsigneeApptID is 9-digits then the content will be returned as one time appointment content (Mail.XML 6.0).
 - If ConsigneeApptID is 5-digits followed by the 'R' character then the content will be returned as recurring content (Mail.XML 6.0).
 - FAST shall only return active/open contents (Mail.XML 4.0 and Mail.XML 6.0)
 - The submitter of a Mail.XML 6.0 DeliveryContentQueryRequest will be denoted by the SubmittingParty SchedulerID or CRID. This replaces the RequestorSchedulerID from Mail.XML 4.0.
 - If both SchedulerID and CRID are provided within the SubmittingParty, they will be validated that they both belong to the same FAST Scheduler (Mail.XML 6.0).
 - FAST shall return only a single content if the ConsigneeContentID and SchedulerContentID fields are present in the DeliveryContentQueryRequest message. In this case, the ConsigneeContentID and SchedulerContentID must correspond (Mail.XML 4.0 and Mail.XML 6.0).
 - The SchedulerID within the header information must be the scheduler of the appointment when ConsigneeAppointmentID is provided and no ConsigneeContentID is provided (Mail.XML 4.0 and Mail.XML 6.0).
 - CreatorSchedulerID must be the content creator when a ConsigneeContentID is provided. FAST shall only return contents where the SubmittingParty is the scheduler of the appointment, content creator, logistics scheduler, mail preparer, or mail owner. The permissions are maintained at the corporate level. A partial scheduler ID (last 5-digits) will always be returned in the DeliveryContentQueryResponse unless the sender of the message is the creator of the content queried or the scheduler of the appointment (when ConsigneeApptID is included (Mail.XML 6.0).
- The DeliveryApptShellUpdateRequest should be used to update one-time and recurring appointment-level detail for an appointment with associated content, however, it may not be used to update the facility of an appointment if the appointment has associated content. The DeliveryApptShellUpdateRequest may only be used to update the facility for one-time and recurring appointments with no associated content.
- The DeliveryApptShellCancelRequest may only be used to cancel one-time appointment shells that have no associated content. The DeliveryApptCancelRequest should be used for any appointments that have associated content.

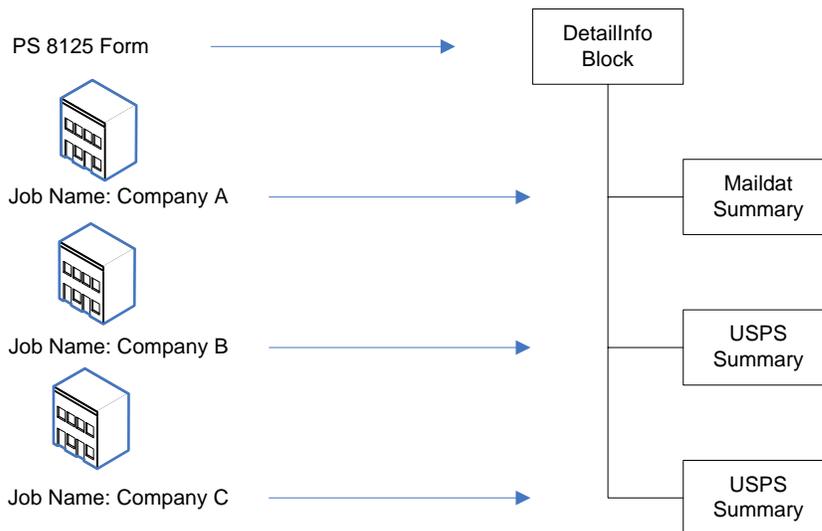
- The DeliveryApptShellCancelRequest may be used to cancel recurring appointments with associated content or recurring appointment shells previously created via Web Services. Additionally, the Mail.XML 6.0 DeliveryApptCancelRequest may be used to cancel recurring appointments by specifying the recurring appointment ID. This applies only to drop ship.
- An appointment created using Mail.XML Version 4.0 may be updated using Mail.XML Version 6.0. An appointment created using Mail.XML Version 6.0 cannot be updated using Mail.XML Version 4.0.
- Content created using Mail.XML Version 4.0 may be updated using Mail.XML Version 6.0. Content created using Mail.XML Version 6.0 cannot be updated using Mail.XML Version 4.0. Content using Mail.XML Version 4.0 cannot be added to an appointment which was last updated using Mail.XML Version 6.0.
- The IDEAlliance specification allows multiple AdvanceMailingID elements to be included in a single USPSSummary block. FAST will only use the first AdvanceMailingID element and others will be ignored.
- The SchedulerContentID in the DeliveryContentCreate, DeliveryContentUpdate and DeliveryContentCancel messages is an ID generated by the Scheduler and should be numeric with no alphabet characters.
- The SchedulerContentID may be populated by the customer with a 12 digit (or less) numeric value. If SchedulerContentID is not populated in a MailDatSummary block, the *PostalOne!* system will populate SchedulerContentID in both the MailDatSummary block and the corresponding USPS Summary block with a *PostalOne!*-generated alphanumeric field consisting of P followed by a sequence of 11 digits or less. The SchedulerContentID will be populated in response messages with this value.
- The FullContentReplacement field in the header of the DeliveryContentUpdateRequest may only be "Yes" if a ConsigneeApptID is provided. If the FullContentReplacement field is "Yes," FAST will cancel all content in an appointment created by the SchedulerID provided. If a ConsigneeContentID is provided in the DeliveryContentUpdateRequest, then the content will be updated and not canceled.
- The TransferAllContent field in the header of the DeliveryApptCancelCreateRequest may only be "Yes" if there is only one appointment that will be cancelled. If the TransferAllContent field is "Yes" FAST will cancel all contents on the appointment
- The PickupOrDelivery element is required in the RequestMsgHeaderInfo, ResponseMsgHeaderInfo and ResponseApptUpdateMsgHeaderInfo. FAST will only support requests messages with "Delivery" as a value for PickupOrDelivery. FAST will ignore "Pickup" as a value for PickupOrDelivery and always treat it as "Delivery".
- The IDEAlliance specification allows for MixedLengthTrays to be a valid palletHandlingUnitType for Origin Entry mail. FAST shall convert all MixedLengthTrays to 2' Trays for Origin Entry mail.
- The ConsigneeApptID, ScheduledInductionDate, LogisticsSchedulerID and LogisticSchedulerCorpID are not required in the DeliveryContentCreateRequest and DeliveryContentUpdateRequest messages. FAST requires either the ConsigneeApptID or all of the following fields be populated: ScheduledInductionDate, LogisticsSchedulerID and LogisticSchedulerCorpID.
- FAST will be supporting First-Class, Standard, Package Services, and Periodicals Mail Classes for Origin Entry appointments in the Mail.XML 6.0 DeliveryApptCreateRequest, DeliveryApptUpdateRequest and DeliveryApptQueryRequest. However FAST will not support a mix of both origin entry and drop ship contents on the same appointment. If DestinationDiscountIndicator is set to 'Yes', the content will be designated as drop ship, if DestinationDiscountIndicator is set to 'No', the content will be designated as origin entry.

- The ContainsAutoMail field will be used to determine whether letter and flats are automated or not for origin entry content.
- The ContainsAutoMail field will only be used for letters and flats.
- The IDEAlliance specification allows for all enumerations of palletType regardless of the Mail Class. FAST requires that the Standard, Package Services, Periodicals and Standard/Periodicals Co-Mailings contents have a palletType of "Pallet". All enumerations of palletType will be accepted for First-Class contents.
- If Customer Supplier Agreement ID (CSAID) is not provided, FAST will return all CSAs associated to the Owner CRID.

Detail Info Block

PostalOne! will support the following method to populate the DetailInfo Block

PostalOne! Option: Shipper uses a DetailInfo Block to represent a PS 8125 Form.
Shipper uses a USPSSummary or MaildatSummary block to represent a mail job.



2.5 Multistop Support

The following is a breakdown of what is supported from a Multistop perspective:

FAST Online Application

- Create a new Appointment and associate it to a new Multistop.
- Create a new Appointment and associate it to an existing Multistop.
- Disassociate an existing Appointment from a Multistop (the Appointment is still active but it is no longer associated with a Multistop).
- Cancel the entire Multistop and therefore cancel all Appointments associated with that Multistop.

Web Services

- Create a new Appointment and associate it to a new Multistop.
- Create a new Appointment and associate it to an existing Multistop.

NOTES:

- An Appointment tied to a Multistop cannot be directly disassociated from that Multistop via Web Services. The Appointment will have to be canceled and rescheduled without the Multistop information OR the Appointment can be disassociated online.
- An existing Appointment cannot be added to an existing or new Multistop through online or Web Services. In both cases, the Appointment will have to be canceled and rescheduled with the Multistop information.
- Any time an Appointment is canceled, the overall Corporate rating may be impacted. Please see Mailer/Appointment Rating section of the *FAST User Guide* for more details on the rating calculation.

2.6 Error Codes/Return Values

The IDEAlliance “TM-Spec-<TM Specification version>Faults” specification outlines a set of possible Return Code values for that Mail.XML specification version. Below is an outline of the Return Code values that are supported in initial version of the USPS implementation. The IDEAlliance specification also allows implementation-specific Return Code values to be defined. The USPS implementation-specific Return Code values are also listed below.

IDEAlliance Return Code Values		
Code	Description	Supported?
1000	Invalid Scheduler ID for Scheduler Corp ID	Yes
1001	Invalid Scheduler Corp ID	Yes
1002	Unrecognized Facility	Yes*
1003	Appt Type not supported by the facility	No
1004	Invalid ConsigneeApptID	Yes
1005	Invalid Multistop ID	Yes
1006	Duplicate Unique Container ID's	Yes
1007	Appt Type not supported by Consignee	No****
1008	Appt Type not supported by facility	No
1009	No slots available for that time / type – dock limit	Yes
1010	The volume requested for the following mail shape exceeded the maximum volume: <Mail Shape>	Yes*
1011	Web Services/Online Appointments must be scheduled at least 1 hour in advance	Yes*
1012	Appointments may only be scheduled up to 14 days in advance	Yes*
1013	Appt type limits exceeded	No
1014	Outside consignee defined timeframes for update/cancel	No**
1015	Update/cancel refused – appointment closed/canceled	Yes
1016	Unrecognized Appointment – field combinations do not match consignee data (generic)	No
1017	Customer and Consignee Appointment ID do not correspond	Yes*
1018	Requested appt date/time is past	Yes
1019	Update/Cancel not allowed, past consignee editable time threshold	No**
1100	Mail.dat User License Code not supported by receiver	Yes

IDEAlliance Return Code Values		
Code	Description	Supported?
1101	Mail.dat Job ID not in consignee database	Yes
1102	Mail.dat DatabaseContainerID not in consignee database	Yes
1103	Mail.dat Verification Zip+4 not in consignee database	Yes
1200	News content is only valid when Mail Class is Periodicals	Yes*
1201	Perishable content is only valid when Mail Class is Package Services	Yes*
1202	Invalid Mail Shape/ Handling Unit combination	Yes
1203	In Home Start Date must be earlier than the In Home End Date	Yes*
1300	Mail Class not supported by Consignee	Yes
1301	Processing Category not supported by Consignee	Yes
1302	Presort Level not supported by Consignee	Yes
1303	Mail Class not supported by Facility	No
1304	Processing Category not supported by Facility	No
1305	Presort Level not supported by Facility	No
1306	Rate Type and Postal class are inconsistent.	Yes
1307	RateType Not Supported by Consignee	Yes ****
1400	Generic Update	No***
1401	Content Property Update	No***
1402	Content Change	No***
1403	ETA Update	No***
1404	Action Requested by Consignee	No***
1405	Delivery Appt Cancelled per request of consignee	Yes
1406	Product combined into another existing appointment	Yes
1407	Major change in expected delivery window	Yes
1408	Product not available	Yes
1409	Appt made in error – human error	Yes
1500	Invalid Tracking ID	Yes
1501	Tracking ID does not Match with Shipper	Yes
1502	Come Back Later	Yes*****
1503	InvalidRequestorSchedulerID for RequestorSchedulerCorpID	Yes
1504	InvalidRequestorSchedulerCorpID	Yes
1505	Appointment Closeout information unavailable	Yes

* - Indicates that the description listed is a more specific USPS description than what is outlined in the IDEAlliance specification.

** - Indicates that this specific code will not be returned but that either the 1011, 1012 or 3010 code will be returned instead, depending on what timeframe rule was violated.

*** - These codes will not be stored or used to implement any business logic in the current USPS implementation.

**** - This specific code will not specifically be supported because the USPS Implementation supports all Appointment types outlined in the IDEAlliance specification. Therefore, if a request is received that contains an invalid Appointment type it will simply fail XSD validation and a Fault block will be returned.

**** - Indicates that this code will be returned if Package Services is sent in. The Rate Type Package Services will be supported by FAST in the future.

***** - This indicates Mailer need to come back again to get the response for the Tracking ID request. This code is returned on TMSpec 1.3 XSD and above.

NOTE: Error Codes 1405-1409 may be provided in the ReturnInfo block of a DeliveryApptCancelRequest, DeliveryApptShellCancelRequest or DeliveryApptContentCancelRequest to indicate the cancellation reason.

PostalOne! – FAST Return Code Values	
Code	Description
3000	Scheduler ID is not valid
3001	Scheduled Appointment time must be scheduled on the hour
3002	Customer and Consignee Multistop ID do not correspond
3003	Perishable content is only valid when Processing Category (Mail Shape) is Machineable Parcel, Outside Parcel (Non-Machineable Parcel), or Irregular Parcel
3004	Optional Appointment Start Time must be before the Optional End Time
3005	Optional Appointment Start Time must be within 8 hours of the Optional End Time
3006	The time difference between the first and last appointment of a multi-stop must be within 24 hours
3007	Appointment date/time must be earlier than In Home Start Date
3008	Number of USPSSummary blocks exceeded maximum limit of 100
3009	A Speedline Appointment cannot have more than 6 pallet positions
3010	Appointments must be updated at least 1 hour in advance
3011	Facility does not accept drop shipments.
3012	An appointment with this Scheduler ID and Scheduler Appointment Request ID already exists
3013	Valid content count must be present for each USPS Summary block
3014	You have experienced an error with the application. Please contact your FAST or Web Services system administrator for help
3015	A Pallet Appointment must have a Count of Pallets Containing Trays, Sacks, Bundles or Parcels
3016	Bedload and Drop and Pick Appointments may not have a Count of Pallets Containing Sacks, Trays, Bundles, or Parcels
3017	A Speedline Appointment may not have a Bedloaded Tray, Sack, Bundle or Parcel Count
3018	<p>Default Facility: Error Message Format: <CIN Group> are not accepted at this facility. Please create an appointment at <NASS Code – Facility Name>.</p> <p>Error Message Example: Standard/Flats are not accepted at this facility. Please create an appointment at 100 – New York.</p> <p>Non-default Facility: Error Message Format: <CIN Group> are not accepted at this facility. Please refer to the Mail Direction Search Report or Drop Ship Product for the correct accepting facility.</p>

PostalOne! – FAST Return Code Values	
Code	Description
	Error Message Example: Standard/Flats are not accepted at this facility. Please refer to the Mail Direction Search Report or Drop Ship Product for the correct accepting facility.
3019	A Speedline Appointment cannot have more than 24 total pallets
3020	Application cannot get response for the given tracking ID (Refer Notes below)
3021	A USPSSummary type was not provided in any of the DetailInfo blocks (See Notes below)
3022	Invalid Mail Preparer ID
3023	Invalid Mail Owner ID
3024	Invalid Publication ID
3027	Update/cancel refused – content does not exist
3028	Invalid Mail Class for Appointment
3029	No update to Content made
3030	Publications only valid when Mail Class is Periodicals
3031	Invalid TM Spec Version
3032	Content may not be cancelled – appointment must have at least one content
3033	Invalid ConsigneeContentID
3034	Scheduler ID does not correspond to Scheduler ID for content
3035	Scheduler ID and Appointment ID do not correspond
3036	Facility ID and Appointment ID do not correspond
3037	Appointment may not be cancelled – appointment has associated content – please use the DeliveryApptCancelRequest message to cancel this appointment
3038	Invalid ConsigneeApptID – Appointment is not a Web Services appointment
3039	RecurringAppt block not supported by Consignee
3040	Update refused – may not change facility for this appointment – this appointment has associated content
3041	Web Services appointments/contents may not be cancelled greater than 24 hours past the appointment date/time
3042	Invalid TM Spec Version – Mailer not certified for TM Spec Version
3043	Content update/cancel refused – content was not created by a DeliveryApptContentCreateRequest message
3044	Invalid TM Spec Version – Appointment has been previously updated with a higher version
3045	Pallet Presort Level Counts must equal Pallet Level Counts
3046	Incorrect WSDL used to send TM2.0 Appointment Message
3047	A Drop and Pick Appointment Type may not be a Multi-Stop
3048	Invalid cancellation reason
3049	Scheduler for appointment must use the DeliveryApptUpdate message to add/update content(s)
3050	Invalid Unique Container ID – Container ID must be 21-digits and begin with 99M
3051	No content associated to appointment
3052	Mailer must provide FAST content ID on update using a DeliveryApptContentUpdateRequest or a DeliveryContentUpdateRequest Message
3053	BMC discount for Periodicals is only valid for cross-docked pallets
3055	Truck lengths <range> <truck length> ft are not accepted at <NASS Code - Facility Name>. Please refer to the Mail Direction Search Report or Drop Ship Product for the correct accepting facility.

PostalOne! – FAST Return Code Values	
Code	Description
3056	Truck lengths <range> <truck length> ft are not accepted at <NASS Code - Facility Name>. Please refer to the Mail Direction Search Report or Drop Ship Product for the correct accepting facility.
3057	No slots available for the requested time/type due to tiering
3058	Number of Pallet Positions must be specified when Appointment Type is Speedline
3059	Facility cannot be modified for recurring appointment instances.
3060	Effective Start Date must be 14 days in the future
3061	Effective Start Date must be before the Effective End Date
3062	Invalid Day Frequency/Week Frequency Combination. A Day Frequency of Daily must have a Week Frequency of Weekly
3063	eVS Indicator is only valid when Processing Category (Mail Shape) is Machineable Parcel, Outside Parcel (Non-Machineable Parcel), or Irregular Parcels
3064	eVS Indicator is only valid when Mail Class is Standard, Package Services or First-Class
3065	eFile Number is only valid when eVS Indicator is Yes
3066	The eFile Number must be 22 digits
3067	Priority Mail Indicator is only valid when Mail Class is First-Class
3068	<Pallet Type> is only valid when the Mail Class is First-Class
3069	Either the appointment ID or all of the following fields are required: scheduled induction date, logistic scheduler ID and logistic scheduler Corp ID
3070	Scheduled Induction Date cannot be in the past
3071	Invalid Logistic Scheduler ID
3072	Invalid Requestor Scheduler ID
3073	Invalid Requestor Scheduler Corp ID
3074	Invalid Requestor Scheduler ID for Requestor Scheduler Corp ID
3075	Invalid TM Spec Version - Content has been previously updated with a higher version
3076	Invalid Creator Scheduler ID for Creator Scheduler Corp ID
3077	Invalid Creator Scheduler Corporate ID
3078	Creator Scheduler ID is not valid
3079	Facility Locale Key is required for this message
3080	Invalid SchedulerContentID/ConsigneeContentID combination
3082	ConsigneeApptID is required to update associated content
3083	Content has been previously associated to an appointment
3084	Invalid SchedulerID – Scheduler ID provided is not the Logistic Scheduler
3085	No more than 100 Contents may be associated to a single appointment
3086	The total volume of Bedload contents may not exceed 9999
3087	The total volume of Pallet, Hamper, APC/ERMC and Wire Container contents may not exceed 9999
3088	Invalid SchedulerContentID
3089	Appointment Time not within Facility hours of operation
3090	The only valid handling units for a First-Class Appointment are 1' Tray, 2' Tray, EMM Tray, Flat Tub, Parcels and Sacks
3091	Consignee does not accept First-Class appointments at this facility
3092	Unable to associate stand-alone content (ConsigneeContentID <Content ID>) - content cancelled
3093	Your search did not return any result

PostalOne! – FAST Return Code Values	
Code	Description
3094	A recurring appointment with this Scheduler ID and Shipper Appointment Request ID already exists.
3095	Effective End Date must not be greater than 12/31/2999
3096	Invalid Recurring Frequency
3098	The Priority Mail indicator is only valid on First-Class Contents
3099	At least 1 Pallet/Bedload count must be provided
3102	A one day In-Home Date Range may not be on a Sunday
3103	Invalid Logistic Scheduler Corp ID
3104	Invalid Logistic Scheduler ID for Logistic Scheduler Corp ID
3105	Full content replacement is only valid when the Consignee Appointment ID is provided
3106	Consignee does not support <Mail Class> for this message type.
3107	Update/Cancel Refused – Content is Closed/Cancelled
3108	Appointment can not contain both drop ship and First-Class contents

NOTE:

- Return code 3018 refers to the Appointment Scheduling and Management Drop Ship Product Validation Process. The new functionality will provide online validation to verify if the selected facility can accept the mail contents on the requested appointment. FAST will utilize the drop ship product ZIP/CIN file. If an appointment passes the CIN group (mail class/mail shape combinations) validation, normal processing will continue as it does today; however, if it fails the CIN group validation, an error message or warning message will be displayed. The error message will notify the user of the type of content(s) that cannot be accepted at the specified facility and which facility those contents should be redirected to for processing. The warning message will notify the user that potential redirections may exist and provides additional facility information for the user to reference. In both cases, the user is provided a hyperlink to the 'facility profile – additional information' page to view specific redirections.
- Return code 3020 is used to indicate the application cannot get a response for this 'TrackingID'. The reason could be the Mailer is using the 'TrackingID' for which they have already obtained a response. This code is supported TMSpec XSD 1.3 and above. This is related to another code, 1502, which indicates the response is not available now and the Mailer needs to come back again later to get a response.
- Return code 3021 is used to indicate that the XML message does not contain any USPSSummary blocks in any of the DetailInfo blocks. USPSSummary blocks provide content information, and appointment create, update, and query messages cannot be processed without content information. Thus, at least one USPSSummary block must be present in at least one DetailInfo block.
- Error Codes 3022 and 3023 are used to validate that the Mail Preparer ID and any Mail Owner IDs are valid FAST Scheduler IDs.
- Error Codes 3025 and 3026 are not included because they are used for Mail.dat Recurring Content Notifications as Error Codes specific to this message type that FAST sends to the PostalOne! system that are.
- Error Code 3031 would be returned in a Fault Response.
- Error Code 3028 is used to indicate whether the mail class for a content is accepted for the appointment. When an appointment shell is created, the Scheduler indicates which Mail Classes are accepted for that appointment. When a DeliveryApptRequest or DeliveryApptUpdateRequest is made, only contents with the same mail class as existing

contents for that appointment may be added using the DeliveryApptContent message request and update types.

- Error Code 3034 is used if a Scheduler who is not associated to the same corporation as the Scheduler who created the content attempts to update or cancel the content.
- Error Codes 3035 and 3036 are used to validate that the Scheduler ID and Facility ID match the Appointment ID given in a Delivery Content Query message.
- Error Code 3037 is returned if a DeliveryApptShellCancelRequest is sent to cancel an appointment that has associated content. The DeliveryApptCancelRequest should be used in this scenario.
- Error Code 3046 Mailers will receive this error messages if they used incorrect WSDL to send TM V2.0 Appointment messages.

PostalOne! – FAST Additional Return Code Values Supported for Mail.XML 6.0	
Code	Description
3109	Invalid Customer Registration ID
3110	Content Cancel Refused - Content is associated to an Appointment
3111	Customer Registration ID is required
3112	Invalid Customer / Supplier Agreement ID for Customer Registration ID
3113	Full Contents Transfer is only applicable if there is only one Appointment that will be cancelled
3114	At least one content must be transferred when canceling multiple appointments
3115	Perishable content is not valid for Origin Entry appointments
3116	Your recurring appointment cannot start on {date} due to constraint limitations at this facility. Please work with the facility to schedule this recurring appointment
3117	The date range may not exceed 7 days
3118	The Start Time may not be in the past
3119	May not search for appointment slots more than 14 days after the current date
3120	An appointment cannot have both drop ship and Origin Entry content
3121	Delivery Content messaging does not support Origin Entry contents
3122	Invalid Customer Registration ID for Scheduler ID
3123	Invalid Partner Scheduler ID
3124	Submitting Party's Scheduler ID is required
3125	Query Refused – {Recurring Appointment Sequence/Unscheduled Appointment/ Appointment Shell} querying is not supported in PartnerApptQueryRequest.
3126	Either the Submitting Party Scheduler ID or Customer Registration ID is required
3127	Invalid Submitting Party Scheduler ID
3128	A Recurring Origin Entry Appointment Sequence must have a Weekly Frequency of "Weekly"
3129	A Rate Type must be selected for Package Services content

2.7 CIN Warning Codes and Descriptions

The following CIN Warnings are returned in the ReturnInfo block of accept messages where they occur. This applies to the following message types:

- DeliveryApptCreateAcceptResponse
- DeliveryApptUpdateAcceptResponse

- DeliveryApptQueryResponse
- DeliveryApptContentCreateAcceptResponse
- DeliveryApptContentUpdateAcceptResponse

PostalOne! – FAST CIN Warnings Supported for Mail.XMLSpec Version 4.0 and 6.0		
Code	Warning Description Text	Warning Explanation
4000	Some < cin group > are accepted at this Facility. Please refer to the Mail Direction Search Report or Drop Ship Product for the correct accepting facility.	If scheduling at a default facility that has a complete redirection, but not all discounts or ZIP Codes (in the case of Delivery Units) are redirected.
4001	Some < cin group > are redirected to < redirect facility >. Please refer to the Mail Direction Search Report or Drop Ship Product for the correct accepting facility.	If scheduling at a redirect facility that has a complete redirection, but not all discounts or ZIP Codes (in the case of Delivery Units) are redirected.
4002	< CIN Group > are being redirected to < redirect facility >. The grace period ends <grace date>.	If scheduling at a default facility that has a complete redirection, but the default grace date of this redirection has not yet expired.
4003	< cin group > processing will no longer be performed at this facility. Please refer to the Mail Direction Search or Drop Ship Product for the correct accepting facility. The grace period ends <grace date>.	If scheduling at a redirect facility that has an expired complete redirection from a default, but that redirection is still in the grace period.
4004	This facility currently accepts <CIN Group>, however, on <redirection start dates>; <CIN Group> will be redirected to <redirect facility>. Please adjust your appointments accordingly.	If scheduling at a default facility that currently has no complete redirections, but a new redirection will become active in 75 days or less
4005	Some < cin group > redirections to this facility expire before < date redirection expires >. Please refer to the Mail Direction Search Report or Drop Ship Product for the correct accepting facility.	If scheduling at a redirect facility that currently has at least one redirection to it, but at least one of those redirections will expire in 75 days or less
4006	Truck lengths <range> <truck length> ft are not accepted at <NASS Code – Facility Name>. If your truck length is <range> <truck length> ft, please create an appointment at <NASS Code – Facility Name>.	If scheduling at a default facility with a truck length that falls within the truck length redirection and the default grace date that has not yet expired.
4007	Truck lengths <range> <truck length> ft are not accepted at <NASS Code – Facility Name>. If your truck length is <range> <truck length> ft, please create an appointment at <NASS Code – Facility Name>.	If an appointment is scheduled at a default facility where a truck length redirection exists and is not within the grace period and no truck length is provided for the appointment.
4008	Truck lengths <range> <truck length> ft are not accepted at <NASS Code – Facility Name>. If your truck length is <range> <truck length> ft, please refer to the Mail Direction Search Report or Drop Ship Product for the correct accepting facility.	If an appointment is scheduled at a redirect facility where a truck length redirection exists and it is within the grace period.

PostalOne! – FAST CIN Warnings Supported for Mail.XMLSpec Version 4.0 and 6.0		
Code	Warning Description Text	Warning Explanation
4009	Truck lengths <range> <truck length>ft are not accepted at <NASS Code – Facility Name>. If your truck length is <range> <truck length> ft, please refer to the Mail Direction Search Report or Drop Ship Product for the correct accepting facility.	If an appointment is scheduled at a redirect facility where a truck length redirection exists and it is not within the grace period and no truck length is provided for the appointment.
4010	BMC discount for Periodicals is only valid for cross-docked pallets.	If an appointment is scheduled for Palletized Periodicals for BMC discount.
4011	< cin group > processing will no longer be performed at this facility. Please refer to the Mail Direction Search or Drop Ship Product for the correct accepting facility. The grace period ends <grace date>.	If an appointment is scheduled at a default facility where the effective end date has expired but is still in the grace period.
4012	The Bill of Lading Number, Shipper Name and Vehicle ID will be updated for the following open appointments that are associated to multi-stop id <multi-stop id>: <appointment id list>.	If a leg of a multi-stop appointment is created or updated, and either the Bill of Lading Number, Shipper Name or Vehicle ID is different from the existing corresponding fields for the multi-stop appointment, then this warning will appear.
4013	Your recurring appointment request has been successfully created. An email notification of the request has been sent to the facility for review. After review, an email response will be sent to you	If a recurring appointment sequence is updated in such a way that a pending request is created (e.g. content or logistics updated on a recurring appointment sequence which is not pure Periodicals.)
4014	Your recurring appointment has been successfully updated. An informational email notification of the approved request has been sent to the facility and the scheduler	If a recurring appointment sequence is updated and auto-approved (e.g. content or logistics updated on a recurring appointment sequence which is pure Periodicals.)
4015	This recurring sequence is pending facility approval	If a pending recurring appointment sequence request is returned in a RecurringApptQueryReponse.
4017	Your recurring appointment has been successfully created. Your recurring appointment currently has no contents. Appointments will not be generated for this Recurring Appointment while there are no contents.	If a recurring appointment sequence is created/updated and there is no associated content.
4018	The requested appointment date and time and in-home date are not in sync with USPS service standards. A shipment with a one day in-home date claiming a BMC discount must be scheduled to arrive by 16:00, five (5) days prior to the specified in-home date.	If standard content has an in-home date range of one day and is being scheduled at a facility that accepts BMC Standard mail.
4019	The requested appointment date and time and in-home date are not in sync with USPS service standards. A shipment with a one day in-home date claiming a SCF discount must be scheduled to arrive by 16:00, three (3) days prior to the specified in-home date	If standard content has an in-home date range of one day and is being scheduled at a facility that accepts SCF Standard mail.

PostalOne! – FAST CIN Warnings Supported for Mail.XMLSpec Version 4.0 and 6.0		
Code	Warning Description Text	Warning Explanation
4020	The requested appointment date and time and in-home dates are not in sync with USPS service standards. A shipment with an in-home date range claiming a BMC discount must be scheduled to arrive by 16:00, five (5) days prior to the second day of the in-home date range.	If standard content has an in-home date range of more than one day and is being scheduled at a facility that accepts BMC Standard mail.
4021	The requested appointment date and time and in-home dates are not in sync with USPS service standards. A shipment with an in-home date range claiming a SCF discount must be scheduled to arrive by 16:00, three (3) days prior to the second day of the in-home date range.	If standard content has an in-home date range of more than one day and is being scheduled at a facility that accepts SCF Standard mail.

The following warnings are returned in the ReturnInfo block of accept messages where they occur. This applies to the following message types:

- DeliveryApptCancelCreateResponse
- DeliveryApptCreateAcceptResponse
- DeliveryApptUpdateAcceptResponse
- DeliveryApptQueryResponse

PostalOne! – FAST Warnings Supported for Mail.XML 6.0		
Code	Warning Description Text	Warning Explanation
4022	Content was not cancelled – it is past the editable time threshold. However, containers for the specified contents have been released for scheduling in other appointments.	If attempting to cancel an appointment and containers less than one hour before the scheduled appointment date/time, or after the scheduled appointment date/time.
4023	Your recurring appointment cannot start on {date} due to constraint limitations. Your recurring appointment will now begin on {date}.	If scheduling at a facility where constraint limitations prevent the instances from beginning on the recurring appointment sequence start date.
4024	In Home Dates are not applicable to Recurring Appointment Series	If including in-home dates when creating/updating a recurring appointment, this warning message will be returned
4025	Appointment <appointment ID> cannot be cancelled, past appointment editable time threshold. Contents/containers of the appointment has been cancelled for scheduling in other appointments.	If attempting to cancel an appointment and containers less than one hour before the scheduled appointment date/time, or after the scheduled appointment date/time.
4026	Your recurring appointment cannot start on {date} due to constraint limitations. Your recurring appointment will now begin on {date}.	If attempting to create or update a recurring appointment and at least the first (and up to the ninth) instances would not be able to be created due to constraint limitations.

Please note that any error or warning messages listed above will be limited to the acceptable field lengths provided in the Mail.XML Specification. This could cause a message to be truncated if it exceeds the acceptable field length.

2.8 Fault Codes

In addition to the above return codes, the IDEAlliance “TM-Spec-1.0Faults” specification also defines a set of possible faults that are used to convey system or operational information. Unlike return codes, faults codes are only three-digit long and they are communicated using the Fault element within the detail section of the SOAP fault. The Mail.XML specification further mandates that the following codes must be understood by all receivers:

Faults	Description
320	Incompatible SOAP Protocol Version
400	Generic Request Error
401	Incomplete/Cannot Parse
402	Not Well Formed XML
403	Validation Failure
404	This Error Intentionally Left Blank
405	Unrecognized Sender
407	Unrecognized Operation
408*	Unrecognized Operation Arguments
410	Not Found
412	Unauthorized
413	Forbidden
430	Not Confirmed
431	Failure Fetching External Data
440	Sorry
500	Generic Internal Responder Error
501	Temporary Responder Problem
503	Not Implemented

* - The fault description will contain additional info about the offending operation argument.

2.9 Web Service Additional Constraints

In addition to conforming to everything stated in the WSDL file, all XML inputs to the Appointment Services operations must qualify the XML elements and attributes with their associated namespace. As an example, following is a good ApptResponseRetrieval input message to the operation RetrieveAppointmentResponses:

ApptResponseRetrievalRequest (Mail.XML 4.0)

```
<maildatTM:ApptResponseRetrievalRequest
xmlns:mailxml="http://idealliance.org/maildat/Specs/md082/tm40/"
xmlns:mailxmlTM="http://idealliance.org/maildat/Specs/md082/tm40/mailxmlTM"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://idealliance.org/maildat/Specs/md082/tm40/mailxmlTM
mailxml_tm_072408.xsd
```

```
http://idealliance.org/maildat/Specs/md082/tm40/
mailxml_base_071408.xsd">
  <maildatTM:ApptResponse maildatTM:TrackingID="1130"/>
</maildatTM:ApptResponseRetrievalRequest>
```

ApptResponseRetrievalRequest (Mail.XML 6.0)

```
<maildatTM:ApptResponseRetrievalRequest
xmlns:mailxml="http://idealliance.org/maildat/Specs/md091/mailxml60a/mail
xml"
xmlns:mailxml_base="http://idealliance.org/maildat/Specs/md091/mailxml60a
/base" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://idealliance.org/maildat/Specs/md091/mailxml60a
/mailxml
mailxml_120408.xsd
http://idealliance.org/maildat/Specs/md091/mailxml60a/base
mailxml_base_120108.xsd">
  <maildatTM:ApptResponse maildatTM:TrackingID="1130"/>
</maildatTM:ApptResponseRetrievalRequest>
```

2.10 Message Response Retrieval Request Processing

The *PostalOne!* system will generate the response to a `MessageResponseRetrievalRequest` (MRRR) for Mail.XML 6.0 requests and `ApptResponseRetrievalRequest` (ARRR) for Mail.XML 4.0 requests.

The *PostalOne!* system will provide the Web Services users with the capability to re-request message responses using the generated Tracking IDs multiple times with a MRRR within 30 days of initial response receipt from FAST. The *PostalOne!* system will provide the users with the latest response from FAST if there are multiple appointment responses for the same TrackingID. The following messages will be supported in the Message Response Retrieval Request processing:

- `CustomerSupplierAgreementQueryResponse`
- `DeliveryApptQueryResponse`
- `DeliveryApptCreateResponse`
- `DeliveryApptCancelResponse`
- `DeliveryApptCancelCreateResponse`
- `DeliveryApptUpdateResponse`
- `DeliveryApptShellCreateResponse`
- `DeliveryApptShellUpdateResponse`
- `DeliveryContentCreateResponse`
- `DeliveryContentUpdateResponse`
- `DeliveryContentCancelResponse`
- `DeliveryContentQueryResponse`
- `Fault`
- `RecurringApptQueryResponse`

- AllApptCloseoutResponse
- PartnerApptQueryResponse

2.11 **PostalOne! WSDL Information**

To support IDEAlliance Mail.XML specification changes and the newly added appointment message types, the *PostalOne!* system will create a new WSDL. Mailers are encouraged to use the new WSDL even if they plan to continue sending and receiving web services requests using previous version of IDEAlliance TM and Mail.XML Appointment requests.

The WSDLs for May Release are available as an appendix to this document and will also be available on RIBBS.

Mailers can access and download the currently supported versions of the *PostalOne!* WSDL from the following location:

<http://www.usps.com/postalone/autodropshipschedule.htm>

2.12 **SchedulerContent ID**

The *PostalOne!* system (Mail.XML Web Services) will generate SchedulerContent ID only if the mailer has set the 'IncludeInSummarizationblock' to 'No'. Mail.XML Web Services will generate the USPS Summary Block and reference Scheduler Content ID. The Scheduler Content ID will be an alphanumeric of the format P###... (P and a sequence number containing up to 11 digits). The prefixing is used to inform FAST that this Scheduler Content ID has been generated by Mail.XML Web Services. Customer will be provided an up to 12-digit number if they are populating this information in the appointment message.

2.13 **Closeout Messages**

FAST will generate a single closeout message for each appointment. The Mail.XML version used to generate the message will be determined as follows:

Closeout messages will be generated by FAST using the latest version used to create or update the appointment. If the latest specification version used to create or update the appointment is no longer supported, the latest supported version for which the scheduler is approved will be used for the closeout message. By the release of the Full-Service program, the latest supported version of Mail.XML is Mail.XML 6.0. For example, if an appointment was created using Mail.XML 4.0, then updated using Mail.XML 6.0, the ConsigneeGoodsReceipt will be generated using Mail.XML 6.0. If an appointment is created/updated using Mail.XML 4.0 and never updated using Mail.XML 6.0, the ConsigneeGoodsReceipt will be generated using Mail.XML 4.0.

Closeout messages for unscheduled appointments will be generated using the latest supported version of Mail.XML for which the scheduler has completed testing. For example, if the scheduler has completed testing for Mail.XML 4.0, the UnscheduledConsigneeGoodsReceipt will be generated using Mail.XML 4.0. If the scheduler has completed testing for Mail.XML 6.0, the UnscheduledConsigneeGoodsReceipt will be generated using Mail.XML 6.0. If the scheduler has completed testing for both Mail.XML 4.0 and Mail.XML 6.0, the UnscheduledConsigneeGoodsReceipt will be generated using Mail.XML 6.0.

The *PostalOne!* system will generate the Closeout Response using the same Mail.XML version as the one used to generate the Closeout Request. A customer who has appointments for both Mail.XML 4.0 and Mail.XML 6.0 will need to send a separate Closeout Request to retrieve closed appointments for Mail.XML 4.0 and Mail.XML 6.0.

The total number of 'consignee goods receipt' and 'unscheduled consignee good receipts' included in a closeout response is currently set to a maximum of 200 messages per response.

2.14 Appointment Shell Cancellation Messages

FAST will support the cancellation of Appointment Shells using both the Delivery Appointment Shell Cancel Request message and Delivery Appointment Cancellation Request Message. Additionally, Appointments with contents can be cancelled using both the Delivery Appointment Shell Cancel Request message and the Delivery Appointment Cancellation Request Message. Recurring appointment shells can only be cancelled using the Delivery Appointment Shell Cancel request.

2.15 *PostalOne!* - FAST Web Services Authentication:

PostalOne! will support new Submitting Party-based security identified in Mail.XML 6.0 for the messages listed below prior to passing on to FAST. *PostalOne!* currently supports these messages with Requestor-based security for Mail.XML 4.0, but these Requestor attributes will be replaced by the Submitting Party block for Mail.XML 6.0. All other messages sent to FAST will continue to support the existing SchedulerID/SchedulerCorpID authentication. FAST shall validate the Submitting Party SchedulerID and CRID (if provided) against FAST Schedulers for the following messages:

- SchedulerID in the Submitting Party block will be authenticated by *PostalOne!* for the DeliveryContentQueryRequest message.
- SchedulerID in the Submitting Party block will be authenticated by *PostalOne!* for the RecurringApptQueryRequest message.
- SchedulerID in the Submitting Party block will be authenticated by *PostalOne!* for the PartnerApptQueryRequest message
- SchedulerID in the Submitting Party block will be authenticated by *PostalOne!* for the CustomerSupplierAgreementQueryRequest message

3.0 **PostalOne! – eDoc Mail.XML Messages**

3.1 **Overview**

This section will discuss the technical information related to the *PostalOne!* - Mail.XML Web Services messages. These messages support all eDocumentation and miscellaneous functionality related to mailer interaction with the *PostalOne!* system in managing, updating and querying mail contents and supplement the Mail.dat functionality, such as payment message queries and container status checking for eDropShip. The *PostalOne!* system will support the new and enhanced version of these messages as described in the external timeline made available to the MTAC 122 work group.

3.2 **eDoc and Miscellaneous Message**

Messages supported by the *PostalOne!* system in late 2009 using Mail.XML 6.0

As part of *PostalOne!* support for the Full-Service program, Mail.XML 6.0 Web Services messages will provide an alternative to and replace the current Wizard Web Services (WWS). Mail.XML will be the overall umbrella specification that will contain all of the current WWS functionality, along with existing Transportation and appointment management as well as extra eDoc messages to support full-service pieces, such as piece detail, piece range, Periodicals reports and postage statements and combined mailing data.

Mail.XML eDOC Web Service Messages will allow business customers to submit and receive postal documents electronically via Web Services. Mail.XML is an alternative to the Mail.dat flat file specification, which works as the whole database of information unlike Mail.XML, which is used for business specific communication.

The following messages will be supported by *PostalOne!* in May 2009:

- MailingGroupQueryRequest
- MailingGroupQueryResponse
- PaymentMessageQueryRequest
- PaymentMessageQueryResponse
- ReconciliationReportQueryRequest
- ReconciliationReportQueryResponse
- ContainerStatusQueryRequest
- ContainerStatusQueryResponse

All eDoc messages have a submitting Party and submitting software.

Submitting Party

Field	Format and Values	Required or Not
MailerID6	Numeric String 6	Optional – Either CRID or MID is required for Data Distribution or eDOC messages
MailerID9	Numeric String 9	Optional – Either CRID or MID is required for Data Distribution or eDOC messages
CRID	String 10	Optional – Either CRID or MID is required for Data Distribution or eDOC messages

SchedulerID	String 12	Optional – Must be used for FAST specific messages
MaildatUserLicense	String 4	Optional – The USPS will use this field for messages that end in the word 'Notification' or 'Delivery' which are PUSH messages and will have a value of 'USPS' in this field for the Push messages from the USPS to the customer.

Submitting Software

Field	Format and Values	Required or Not
SoftwareName	String	Required
Vendor	String	Required
Version	String	Required
ApprovalDate	Date Time	Optional – USPS will require this date
ApprovalKey	String	Optional – USPS will require this key given out by the USPS help desk to the software vendor/IT shops to verify the approval process

3.3 Mail.XML Specific Message Details

Mailing Group Query Request and Response

The Mailing Group Query Request and Response messages allow the customer to provide a Customer Group ID (Customer Job ID) to open a mailing group and in response the USPS *PostalOne!* system sends back all the basic information about the mailer, including Permit or Publication Number information as well as the mailing acceptance office ZIP Code and the type of presort used in the mailing while assigning a MailingGroupID.

For more details on required vs. optional fields and blocks as well as the data types, please review the Mail.XML Transaction Message Protocol Messaging Specification.

PaymentMessageQueryRequest and Response

After paying for postage, send the PaymentMessageQueryRequest. The request requires the Statement ID, tracking ID, mailing and Customer group ID besides the submitting party and submitting software.

PaymentMessageQueryResponse

PaymentMessageQueryResponse includes a QueryReponse and a QueryError block. The QueryResponse includes Postage statement data blocks for postage statement(s). Each Postage statement data block includes PostageStatementTransactionNumber, CustomerReferenceTransactionNumber, MailingFacility, PostageAmountPaid, DateOfTransaction, and TimeOfTransaction. Both messages include ReturnInfo.

Reconciliation Report Query Request and Response

The customer cannot update the Qualification report once filed. After the customer goes through the process of printing and delivering the mail to the USPS, the USPS creates a reconciliation report that shows how much mail was planned and how much was mailed. Such reports are called Reconciliation Reports.

In the ReconciliationReportQueryRequest, the customer is required to provide the mailing group ID and Customer group ID besides the mandatory elements of Submitting party and Submitting software. In the ReconciliationReportQueryResponse message the *PostalOne!* system responds back with a reconciliation report for every qualification report with in the Job/ mailing group. The USPS responds back with one Mailing Group Summary and one or more Reconciliation Summary reports. The message also includes an optional Return Code and description. If there are issues, the USPS responds back with the Query Error block and provides one or more Return codes and descriptions to explain the errors.

Container Status Messages

The container status message set for PS Form 8017 and PS form 8125 containers allows a mailer, mail owner, or consolidator to provide container key information and find out if the container has any eDropShip-related or any other issues prior to shipping the container to the USPS. The following eDropShip issues can stop the creation of an electronic 8125 (e8125) are:

- invalid barcode – the barcode data format is not valid
- payment not finalized – the clerk has not yet finalized payment
- payment not yet processed – the electronic data has not been updated by the mailer to make the container in a ready to pay status
- entry point payment discount error – the entry point discount filed vs. the physical facility where mail is scheduled through FAST for delivery does not support the discount type

In all of these cases, unless the issues are resolved electronically through an update of data or through the clerk finalizing the postage, the containers' e8125 will not be generated and sent to the Surface Visibility (SV) devices for induction. When that container shows up without an e8125 at the plant, the container can be stopped and its processing can be delayed or it can be returned to the shipper.

This message set includes:

- Container Status Query Request
- Container Status Query Response

3.4 Message Response Retrieval Request 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 Message Response Retrieval Request/Response message set which includes:

- Message Response Retrieval Request
- Message Response Retrieval Response

In Full-Service Release 1, customers will have the ability to provide tracking ID in the MessageResponseRetrievalRequest message to retrieve the following responses:

- ContainerStatusQueryResponse
- MailingGroupQueryResponse
- PaymentMessageQueryResponse

- ReconciliationReportQueryResponse

3.5 Additional Rules/Constraints

Container Status Messages

The *PostalOne!* system will support status checking through Container Status Query Request and Response messages in this release by the originator of the data and by the entity identified as the Scheduler or Carrier in the eDocumentation. Status checking allows mailers to view the status of any given container(s) and ensure when the container is shipped that the induction facility will not have any issues in accepting the containers.

- The *PostalOne!* system upon receiving the container status message shall go to the container management system of eDropShip to find out all validation status of the requested container(s) and respond to the customer with the status information.
- If there are any issues or validation errors for one or more containers, then the *PostalOne!* system shall respond back to the customer with validation failures for one or more container(s).

3.6 Error Codes/Return Values

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 consists of a tracking ID and one or more fault codes and optional fault descriptions.

Below is the list of the error codes that will be used by USPS when deploying the eDocumentation Messages in May 2009

PostalOne! – eDocumentation Return Code Values	
Code	Description
5000	FULLSERVICE-EDOC: REQUEST SERVED SUCCESSFULLY.
5001	FULLSERVICE-EDOC: Not Valid Schema.
5002	FULLSERVICE-EDOC: The Customer Group ID is required.
5003	FULLSERVICE-EDOC: The Customer Group ID contains illegal meta-characters.
5004	FULLSERVICE-EDOC: Mailing Group does not exist with Customer Group ID.
5005	FULLSERVICE-EDOC: User does not have access to the mailing group's permit.
5006	FULLSERVICE-EDOC: Internal data error message Please Contact <i>PostalOne!</i> Help Desk.
5007	FULLSERVICE-EDOC: Internal System error message Please Contact <i>PostalOne!</i> Help Desk.
5008	FULLSERVICE-EDOC: Empty Message. Please Contact <i>PostalOne!</i> Help Desk.
5009	FULLSERVICE-EDOC: Request is not XML Message. Please Contact <i>PostalOne!</i> Help Desk.
5010	FULLSERVICE-EDOC: NO DATA FOUND.
5501	Barcode Validation Error: Invalid Barcode Length
5502	Barcode Validation Error: Invalid Barcode Prefix
5503	Barcode Validation Error: Invalid Barcode
5504	Entry Point Validation Error: Invalid Entry Point Discount Type
5505	Entry Point Validation Error: Invalid Entry Point Discount Location
5506	Appointment Validation Error: Container associated with multiple appointments
5507	Payment Validation Error: No postage statements associated with the container
5509	Barcode Validation Error: Missing Barcode

PostalOne! – eDocumentation Return Code Values	
Code	Description
5508	Unknown Error
5510	Barcode Validation Error: Duplicate Barcode
5511	Payment Validation Error: Container Payment Not Finalized
5512	No matching records were found for the given criteria
5900	Mail.XML message received is not supported(zzz ->MessageName)
5901	Mail XML version is not supported

3.7 Additional Constraints

TBD

3.8 WSDL Information

The WSDLs for Release 1 are available as an appendix to this document and will be available on RIBBS.

Mailers can access and download the currently supported version of the *PostalOne!* WSDL from the following location:

<http://www.usps.com/postalone/autodropshipschedule.htm>

4.0 Full-Service Data Distribution – Release 1

4.1 Purpose

This section describes the technical implementation and ways in which customers will be able to receive the full-service data feedback as part of the USPS Full-Service implementation.

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

4.2 Which methods does the *PostalOne!* system support for full-service data feedback?

The USPS will provide multiple options to receive the full-service data, which are as follows:

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	No	Yes	Yes	Yes
2	Container Visibility/Scan	No	Yes	Yes	Yes
3	ACS COA	Yes	No	Yes	Yes
4	ACS Nixie Detail	Yes	No	Yes	Yes

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 the FTP protocol to its customers will provide ample time and communication to the USPS customers and collaborate with the customers to help with transition before transitioning to the non-FTP protocol data communication methods.

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

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.

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 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.

4.3 What do you need to receive your data through the data exchange methods identified?

Step 1

The customers must be registered *PostalOne!* users and must have signed data exchange agreements in order to send or receive electronic data from the *PostalOne!* system.

It is as easy as visiting the *PostalOne!* home page at www.usps.com/postalone and clicking the **Learn More** button on the right under New Customers. There, you will find additional information about the program and application process. If you have questions, call the *PostalOne!* Customer Care Center 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 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 is version 6.0-B 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 will publish and provide 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 automatically receive either the detail data or the notifications as they pertain to StartTheClock, Container Scans, ACS COA, and 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.

4.4 How do you implement/use the Mail.XML available for full-service data distribution?

Available at <http://www.mailxml.org>.

Why Mail.XML-based Data Distribution?

Mail.XML version 6.0-B 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 easily and efficiently 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. 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.

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 *PostalOne!* users. The customer must have gone through the *PostalOne!* approval and activation process for this business function and must establish push subscription profiles in the *PostalOne!* system 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.

4.5 Messages Supported in Release 1

The following messages are supported in Release 1 of *PostalOne!* Full-Service:

FullServiceAddressCorrectionDelivery	CastofCharactersCancelRequest
FullServiceAddressCorrectionNotification	CastofCharactersCancelResponse
FullServiceAddressCorrectionQueryRequest	CastofCharactersCreateRequest
FullServiceAddressCorrectionQueryResponse	CastofCharactersCreateResponse
FullServiceContainerVisibilityDelivery	CastofCharactersUpdateRequest
FullServiceContainerVisibilityNotification	CastofCharactersUpdateResponse
FullServiceContainerVisibilityQueryRequest	DeliveryResponse
FullServiceContainerVisibilityQueryResponse	NotificationResponse
FullServiceNixieDetailDelivery	
FullServiceNixieDetailRequest	
FullServiceNixieDetailResponse	
FullServiceNixieNotification	
StartTheClockDelivery	
StartTheClockNotification	
StartTheClockQueryRequest	
StartTheClockQueryResponse	

4.6 Push Messages

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. 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 *PostalOne!* 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."

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.

4.7 Pull Messages

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.

4.8 Messages

The USPS will initially implement the following Mail.XML messages in support of full-service Push/Pull methods for full-service data feedback.

Push Messages Supported in Release 1 of Full-Service Implementation:

Response Messages for Notification and Delivery

- NotificationResponse
- DeliveryResponse

StartTheClock :

- StartTheClockNotification
- StartTheClockDelivery

Container Visibility :

- FullServiceContainerVisibilityNotification
- FullServiceContainerVisibilityDelivery

ACS COA:

- FullServiceAddressCorrectionNotification

- FullServiceAddressCorrectionDelivery

ACS Nixie:

- FullServiceNixieDetailNotification
- FullServiceNixieDetailDelivery

For the above messages, you must implement the USPS-provided WSDL and XSD and to implement on your side as a server solution that can accept requests from the USPS *PostalOne!* system and to provide the proper URL and access information necessary for the *PostalOne!* system to push the data to your server.

Pull Messages Supported in Release 1 of Full-Service Implementation:

StartTheClock :

- StartTheClockQueryRequest
- StartTheClockQueryResponse

Container Visibility :

- FullServiceContainerVisibilityQueryRequest
- FullServiceContainerVisibilityQueryResponse

ACS COA:

- FullServiceAddressCorrectionQueryRequest
- FullServiceAddressCorrectionQueryResponse

ACS Nixie:

- FullServiceNixieDetailQueryRequest
- FullServiceNixieDetailResponse

Cast Of Characters Messages for Release 1 of Full-Service Implementation:

CastOfCharacters:

- CastOfCharactersCreateRequest
- CastOfCharactersCreateResponse
- CastOfCharactersCancelRequest
- CastOfCharactersCancelResponse
- CastOfCharactersUpdateRequest
- CastOfCharactersUpdateResponse

4.9 Message Response Retrieval Request 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 which includes:

- Message Response Retrieval Request
- Message Response Retrieval Response

In Full-Service Release 1, customers will have the ability to provide tracking ID in the MessageResponseRetrievalRequest message to retrieve the following responses:

- CastofCharactersCreateResponse
- CastofCharactersCancelResponse
- CastofCharactersUpdateResponse
- FullServiceAddressCorrectionQueryResponse
- FullServiceNixieDetailQueryResponse
- FullServiceContainerVisibilityQueryResponse
- StartTheClockQueryResponse

4.10 Additional Business Rules

Delivery and Notification Messages:

Response Messages for Notification and Delivery

- NotificationResponse
 1. Every time the USPS sends a Push “Notification” message to the customer, if the customer has registered for one or more “Notification” messages, the *PostalOne!* system expects a “Notification Response” message from the customer to let the *PostalOne!* system know if everything is OK.
 2. The return codes and their descriptions for the Response messages are published in section 4.12 of this guide. They will be updated by the USPS for implementation based upon the schedule made available in the External Timeline for IM implementation document.
- DeliveryResponse
 1. Every time the USPS sends a Push ‘Delivery’ message to the customer, if the customer has registered for one or more ‘Delivery’ messages, the USPS *PostalOne!* program expects a ‘Delivery Response’ message from the customer to let the *PostalOne!* system know if everything is OK. 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.
 2. 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. The return codes and their description are provided in section 4.12 of this guide.

StartTheClock:

- StartTheClockNotification

With each StartTheClockNotification message, 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.

- StartTheClockDelivery

With each StartTheClockDelivery message, 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.

Container Visibility:

- FullServiceContainerVisibilityNotification

With each FullServiceContainerVisibilityNotification message, 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 to the customer, so the customer can use the Job information in his/her Query request.

- FullServiceContainerVisibilityDelivery

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

- OPTIONAL LargeTransactionDivideroptionalGroupType. 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.
 - 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 a count of records in the current message transaction.
- e. TotalRecordsAcrossMessages – This is the total expected record count when the receiver for the MessageGroupID has received all transactions.
- f. 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
 - 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
- 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 FullServiceContainerVisibility blocks.

ACS COA:

- FullServiceAddressCorrectionNotification

With each FullServiceAddressCorrectionNotification 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.

- FullServiceAddressCorrectionDelivery

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

1. OPTIONAL LargeTransactionDivideroptionalGroupType. 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. TotalRecordsAccrossMessages – 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 FullServiceAddressCorrection 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:
 - Individual/Family Move Type = Last Name, First Name, Prefix and Suffix
 - Business Move Type = Last Name 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:
 - 'A' = Package Services (Bound Printed Matter)
 - 'B' = First-Class
 - 'C' = Periodicals Initial

- 'D' = Standard Mail
- '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:
 - <blank> = COA Match with new address
 - 'K' = COA Match, no forwarding address
 - 'G' = COA Match, PO Box Closed, no forwarding address
 - '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:
 - 'G' = General Delivery
 - 'H' = Highway Contract Route
 - 'P' = Post Office Box
 - 'R' = Rural Route
 - 'S' = Street Address

ACS Nixie:

- FullServiceNixieDetailNotification

With each FullServiceNixieDetailNotification 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.

- FullServiceNixieDetailDelivery

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

1. OPTIONAL LargeTransactionDivideroptionalGroupType. 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. TotalRecordsAccrossMessages – 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:
 - 'F' = Forwarded
 - 'W' = Wasted
 - 'R' = Returned
 - 'U' = Unable to determine action
 - 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:
 - 'A' = Package Services (Bound Printed Matter)
 - 'B' = First-Class
 - 'C' = Periodicals Initial
 - 'D' = Standard Mail
 - '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:
 - 'A' = Attempted, not known
 - 'B' = Returned for better address
 - 'D' = Outside delivery limits

- 'E' = In dispute
- 'I' = Insufficient Address
- 'L' = Illegible
- 'M' = No mail receptacle
- 'N' = No such number
- 'P' = Deceased
- 'Q' = Not deliverable as addressed/unable to forward
- 'R' = Refused
- 'S' = No such street
- 'U' = Unclaimed
- 'V' = Vacant
- 'X' = No such office

Request and Response Messages

StartTheClock:

- StartTheClockQueryRequest

With each StartTheClockQueryRequest 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 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.

- StartTheClockQueryResponse

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 info 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 info and the Customer Group ID and MailingGroupID represent the Mail.XML/Postal Wizard info.
 - 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.

Container Visibility:

- FullServiceContainerVisibilityQueryRequest

With each FullServiceContainerVisibilityQueryRequest 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 info or Mail.dat Job info 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 info or can optionally provide the Mail.dat ContainerID in the ContainerID element.

- FullServiceContainerVisibilityQueryResponse

With each FullServiceContainerVisibilityQueryResponse 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 FullServiceContainerVisibilityQueryResponse has two major choice blocks one is QueryResults and the other is QueryErrors.

- In QueryResults,
 - The job info 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 info 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 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.

ACS COA:

- FullServiceAddressCorrectionQueryRequest

With each FullServiceAddressCorrectionQueryRequest 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 OwingParty 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.
- FullServiceAddressCorrectionQueryResponse

With each FullServiceAddressCorrectionQueryResponse 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 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 FullServiceContainerVisibilityQueryResponse has two major choice blocks one is QueryResults and the other is QueryErrors.

- In QueryResults,
 - The customer will receive zero or more FullServiceAddressCorrection blocks with job info and address correction 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.

ACS Nixie:

- FullServiceNixieDetailQueryRequest

With each FullServiceNixieDetailQueryRequest 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 elementA valid software vendor company name in the vendor element

- b. The most current version number of the software in the Version element
 - c. 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 info.
- FullServiceNixieDetailResponse

With each FullServiceNixieDetailResponse 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 FullServiceNixieDetailResponse 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 info 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.

Cast Of Characters Messages for Release 1 of Full-Service Implementation:

CastOfCharacters:

The Cast Of Characters messages are used to distribute data to a third party by a Mail Preparer or Mail Owner. All of the full-service data for StartTheClock, Container Scans, ACS COA, ACS Nixie, and Confirm can be diverted to another party, if that third party is pre-identified in the *PostalOne!* 'Dynamic' profile.

- CastOfCharactersCreateRequest

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

1. The customer must provide an internal Tracking ID in the header attribute of OriginalSenderMsgID
2. 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.
3. 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.
4. The customer must provide one or more MailPieceGroup blocks on the type of data that needs to be distributed.
5. The customer may provide one or more MailPieceGroup blocks and provide either the IMB info or provide the IMB lowest or highest ID block information.
6. The customer must identify the business entity who is to receive the data in the Participant block by only using the MID of the participant.
7. The customer must identify the role in the Role element type to define what type of data should be sent to the participant. Valid data types are ACS subscriber, ACS Data Receiver, Confirm Subscriber, Confirm Data Receiver, ContainerVisibility, and StartTheClock.
8. The customer must identify the date on which the Cast Of Characters direction to distribute data to third party becomes active and valid.
9. The customer may provide an optional EffectiveEndDate, duration, or DurationUnitOfMeasure.

- CastOfCharactersCreateResponse

With each CastOfCharactersCreateResponse message, the *PostalOne!* system will provide the Mandatory OriginalSenderMsgID and the ReceiverMessageID.

- CastOfCharactersCreateAccept or CastOfCharactersReject blocks.

1. In CastOfCharactersCreateAccept block, the customer will receive confirmation group ids for the MailPieceGroup with a return code and description. The return Codes and descriptions will be provided by the USPS based upon the schedule made available in the External Timeline for IM implementation document.
2. In CastOfCharactersCreateReject block,
 - a. The customer will receive the return Codes and descriptions, which will be provided by the USPS in the Timeline for IM implementation document.
 - b. With the return Code and descriptions, the USPS may provide the details back to the customer on the piece or Container data provided to USPS, which was rejected.

- CastOfCharactersCancelRequest

The CastOfCharactersCancelRequest can be used to cancel one or more records that the customer previously sent to the USPS. The following fields can be used to cancel all or some of the records.

The customer can provide the following information in blocks.

1. The customer must provide an internal Tracking ID in the header attribute of OriginalSenderMsgID
 2. 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.
 3. SubmittingSoftware – The *PostalOne!* system for messages that it receives for the Mail.XML will always require
 - A valid software name in the SoftwareName element
 - A valid software vendor company name in the vendor element
 - The most current version number of the software in the Version element
 - 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.
 4. The customer can provide optional one or more MailPieceGroup blocks on the type of data that needs to be distributed.
 5. The customer may provide one or more MailPieceGroup blocks and provide highest ID block information.
 6. The customer may identify the business entity who is to receive the data in the Participant block by only using the MID of the participant.
 7. The customer may identify the role in the Role element type to define what type of data should be sent to the participant. Valid data types are ACS subscriber, ACS Data Receiver, Confirm Subscriber, Confirm Data Receiver, ContainerVisibility, and StartTheClock.
- CastOfCharactersCancelResponse

With each CastOfCharactersCancelResponse message, the *PostalOne!* system will provide the Mandatory OriginalSenderMsgID and the ReceiverMessageID.

1. CastOfCharactersCancelAccept or CastOfCharactersCancelReject blocks.
 - a. In CastOfCharactersCancelAccept block, the customer will receive confirmation group ids for the MailPieceGroup or for the ContainerGroup Blocks
 - b. The received participant block by using the MID of the participant will be provided.
 - c. Cancel Confirmation ID number will be provided.
 - d. Return info block with be return code and description. The return Codes and descriptions will be provided by the USPS based upon the schedule made available in the External Timeline for IM implementation document.
 - e. In CastOfCharactersCancelReject block,
 - The customer will receive the return Codes and descriptions, which will be provided by the USPS in the Timeline for IM implementation document.
 - With the return Code and descriptions, the USPS may provide the details back to the customer on the piece or Container data provided to USPS, which was rejected.
- CastOfCharactersUpdateRequest

The CastOfCharactersUpdateRequest can be used to update one or more records that the customer previously sent to the USPS. The Cast of Characters update request is used to modify existing data recipient information – Roles, EffectiveDate, Expiration date, Duration, Duration Unit, and MailPieceGroupID.

The following message fields can be used to update all or some of the records:

1. The customer must provide an internal Tracking ID in the header attribute of OriginalSenderMsgID
 2. 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.
 3. SubmittingSoftware – The *PostalOne!* system for messages that it receives for the Mail.XML will always require
 - A valid software name in the SoftwareName element
 - A valid software vendor company name in the vendor element
 - The most current version number of the software in the Version element
 - 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.
 4. The customer must provide one or more MailPieceGroup blocks on the type of data that needs to be distributed.
 5. The customer may provide one or more MailPieceGroup blocks and provide either the IMB info or provide the IMB lowest or highest ID block information.
 6. The customer may identify the business entity who is to receive the data in the Participant block by only using the MID of the participant.
 7. The customer may identify the role in the Role element type to define what type of data should be sent to the participant. Valid data types are ACS subscriber, ACS Data Receiver, Confirm Subscriber, Confirm Data Receiver, ContainerVisibility, and StartTheClock.
 8. The customer may identify the date on which the Cast Of Characters direction to distribute data to third party becomes active and valid.
 9. The customer may provide an optional EffectiveEndDate, duration, or DurationUnitOfMeasure.
- CastOfCharactersUpdateResponse

With each CastOfCharactersUpdateResponse message, the *PostalOne!* system will provide the Mandatory OriginalSenderMsgID and the ReceiverMessageID.

For the CastOfCharactersUpdateAccept or CastOfCharactersUpdateReject blocks:

1. In CastOfCharactersUpdateAccept block, the customer will receive confirmation group ids for the MailPieceGroup Blocks
2. The received participant block by using the MID of the participant will be provided.
3. Return info block with be return code and description. The return Codes and descriptions will be provided by the USPS based upon the schedule made available in the External Timeline for IM implementation document.
4. In CastOfCharactersUpdateReject block,
 - The customer will receive the return Codes and descriptions, which will be provided by the USPS in the Timeline for IM implementation document.

- With the return Code and descriptions, the USPS may provide the details back to the customer on the piece or Container data provided to USPS, which was rejected.

4.11 Pull Request/Response-Handling Large Transactions

The following messages can get large transactions as part of response. The Mail.XML specification has defined an element 'LargeTransactionDividerResult' to support such transactions. The *PostalOne!* system will divide the large result set into smaller chunks. On the response provided it would include TrackingID as part of 'LargeTransactionDividerResult'. Whenever a mailer application sees this element, they need to send MessageResponseRetrievalRequest (MailXML6.0) or ApptResponseRetrievalRequest (TM2.0/MailXML4.0) and use the TrackingID provided to get the response.

For example, a mailer sends 'FullServiceNixieDetailQueryRequest' 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 of first 200 messages and include five TrackingID as part of the 'LargeTransactionDividerResult'.

Mailer can send TrackingID request and get the remaining data. There is also element 'LargeTransactionDividerGroupType' which gives more information on what is included on each transmission.

- FullServiceAddressCorrectionQueryResponse
- FullServiceContainerVisibilityQueryResponse
- FullServiceNixieDetailQueryResponse
- Start the Clock Query Response

4.12 Error Codes/Return Values

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.

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

PostalOne! – Data Distribution- Delivery Response Return Code Values	
Code	Description
5800	Successful Push Delivery Receipt
5801	Push Authentication failed: Invalid Username
5802	Push Authentication failed: Incorrect Password
5803	Push Account Disabled
5804	Push Delivery Processing Failed

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

PostalOne! – Data Distribution Return Code Values	
Code	Description

PostalOne! – Data Distribution Return Code Values	
Code	Description
5805	Push Subscription Error: Invalid CRID.
5806	Push Subscription Error: Invalid message type.
5807	Push Subscription Error: Invalid Push type.
5808	Push Subscription Error: Invalid user ID and/or password.
5809	The web service URL format is invalid.
5810	Push Subscription Error: Push Start date is invalid.
5811	Push Subscription Error: Invalid frequency.
5812	Push Subscription created successfully.
5813	Push Subscription updated successfully.
5814	Push Subscription cancelled successfully.
5815	Push Subscription Error: Database error occurred, subscription not saved/updated.
5816	Push Subscription Error: Database error occurred, subscription already exists.
5817	Push Subscription Error: Database error occurred, subscription does not exist.

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:

PostalOne! - Cast of Characters Response Return Code Values	
Code	Description
6000	Success
6001	Invalid Submitting Mailer ID
6002	Invalid Participant Mailer ID
6003	Invalid Submitting CRID ID
6004	For Future Use
6005	Submitting Software Version is not supported
6006	Unknown Error
6007	Mail XML version is not supported
6008	Mailing Piece Group Range does not exist
6009	For Future Use
6010	Can not update record, original record does not exist
6011	Can not cancel record, original record does not exist
6012	XML Not well formed
6013	Schema Validation failure
6014	Data distribution profile has not been established by the mailer
6015	Invalid Mailing Piece Group Range which overlaps with an existing profile
6016	Can not update record, mailpiece group has not been defined previously
6017	Can not perform request, mailer is blocked

Appendix A - WSDLs and XSDs

WSDL for the Release 1 – Also will be available as a downloadable attachment from RIBBS

```
<?xml version="1.0" encoding="UTF-8"?>
<!--      Date      Comments      -->
<!--      11/17/2007      MailXML -POAppointmentServices WSDL      Initial
version 1.0 -->
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://www.usps.com/postalone/services/POAppointmentServices20"
xmlns:auth="http://www.usps.com/postalone/services/UserAuthenticationSche
ma"
xmlns:pdd="http://www.usps.com/postalone/services/PushUserProfileSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.usps.com/postalone/services/POAppointmentServices20"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
name="POAppointmentServices20"
targetNamespace="http://www.usps.com/postalone/services/POAppointmentServ
ices20">
    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Version
1.0.0</wsdl:documentation>
    <wsdl:types>
        <xsd:schema
targetNamespace="http://www.usps.com/postalone/services/POAppointmentServ
ices20"
xmlns="http://www.usps.com/postalone/services/POAppointmentServices20"
xmlns:auth="http://www.usps.com/postalone/services/UserAuthenticationSche
ma"
xmlns:pdd="http://www.usps.com/postalone/services/PushUserProfileSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.usps.com/postalone/services/POAppointmentServices20"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:import
namespace="http://www.usps.com/postalone/services/UserAuthenticationSchem
a" schemaLocation="UserAuthenticationInfo.xsd"/>
            <xsd:import
namespace="http://www.usps.com/postalone/services/PushUserProfileSchema"
schemaLocation="PushUserProfileInfo.xsd"/>
            <!--      Appointment Messages      -->
            <xsd:element name="apptQuery" type="xsd:string"/>
            <xsd:element name="apptRequest" type="xsd:string"/>
            <xsd:element name="apptUpdate" type="xsd:string"/>
            <xsd:element name="apptCancel" type="xsd:string"/>
            <xsd:element name="apptCloseout" type="xsd:string"/>
            <xsd:element name="apptResponseRetrieval"
type="xsd:string"/>
        </xsd:schema>
    </wsdl:types>
</wsdl:definitions>
```

```

type="xsd:string" />
    <xsd:element name="recurringApptQuery"
type="xsd:string" />
    <xsd:element name="messageResponseRetrieval"
type="xsd:string" />
    <xsd:element name="apptCancelCreate"
type="xsd:string" />
    <xsd:element name="apptStatus" type="xsd:string" />
    <xsd:element name="result" type="xsd:string" />
    <xsd:element name="fault" type="xsd:string" />
    <!-- Appointment Shell Messages -->
    <xsd:element name="apptShellUpdate" type="xsd:string" />
    <xsd:element name="apptShellCancel" type="xsd:string" />
    <!-- Content Messages -->
    <xsd:element name="apptContentCreate"
type="xsd:string" />
    <xsd:element name="apptContentUpdate"
type="xsd:string" />
    <xsd:element name="apptContentCancel"
type="xsd:string" />
    <xsd:element name="contentQuery" type="xsd:string" />
    <xsd:element name="apptShell" type="xsd:string" />
    <!-- eDoc Messages-->
    <xsd:element name="mailingGroupQuery"
type="xsd:string" />
    <xsd:element name="paymentMessageQuery"
type="xsd:string" />
    <xsd:element name="reconciliationReportQuery"
type="xsd:string" />
    <xsd:element name="containerStatusQuery"
type="xsd:string" />
    <xsd:element name="partnerApptQuery"
type="xsd:string" />
    <xsd:element name="customerSupplierAgreementQuery"
type="xsd:string" />
    <!-- Data Distribution Service Measurement(Pull
Messages)-->
    <xsd:element name="fullServiceAddressCorrectionQuery"
type="xsd:string" />
    <xsd:element name="fullServiceContainerVisibilityQuery"
type="xsd:string" />
    <xsd:element name="fullServiceNixieDetailQuery"
type="xsd:string" />
    <xsd:element name="startTheClockQuery"
type="xsd:string" />

```

```

        <xsd:element name="byForConflictQuery"
type="xsd:string" />
        <xsd:element name="castofCharactersCreate"
type="xsd:string" />
        <xsd:element name="castofCharactersUpdate"
type="xsd:string" />
        <xsd:element name="castofCharactersCancel"
type="xsd:string" />
        <xsd:element name="deliveryPointValidationQuery"
type="xsd:string" />
        <xsd:element name="pushSubscriptionCreate"
type="xsd:string" />
        <xsd:element name="pushSubscriptionUpdate"
type="xsd:string" />
        <xsd:element name="pushSubscriptionCancel"
type="xsd:string" />
    </xsd:schema>
</wsdl:types>
<wsdl:message name="DeliveryApptQueryRequest">
    <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
    <wsdl:part element="tns:apptQuery" name="apptQuery" />
</wsdl:message>
<wsdl:message name="DeliveryApptRequest">
    <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
    <wsdl:part element="tns:apptRequest" name="apptRequest" />
</wsdl:message>
<wsdl:message name="AllDeliveryApptCloseoutRequest">
    <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
    <wsdl:part element="tns:apptCloseout" name="apptCloseout" />
</wsdl:message>
<wsdl:message name="DeliveryApptStatusRequest">
    <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
    <wsdl:part element="tns:apptStatus" name="apptStatus" />
</wsdl:message>
<wsdl:message name="RecurringApptQueryRequest">
    <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
    <wsdl:part element="tns:recurringApptQuery"
name="recurringApptQuery" />
</wsdl:message>

```

```

        <wsdl:message name="DeliveryApptContentUpdateRequest">
            <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
            <wsdl:part element="tns:apptContentUpdate"
name="apptContentUpdate"/>
        </wsdl:message>
        <wsdl:message name="ApptResponseRetrieval">
            <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
            <wsdl:part element="tns:apptResponseRetrieval"
name="apptResponseRetrieval"/>
        </wsdl:message>
        <wsdl:message name="DeliveryContentQueryRequest">
            <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
            <wsdl:part element="tns:contentQuery" name="contentQuery"/>
        </wsdl:message>
        <wsdl:message name="DeliveryApptContentCancelRequest">
            <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
            <wsdl:part element="tns:apptContentCancel"
name="apptContentCancel"/>
        </wsdl:message>
        <wsdl:message name="DeliveryApptContentCreateRequest">
            <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
            <wsdl:part element="tns:apptContentCreate"
name="apptContentCreate"/>
        </wsdl:message>
        <wsdl:message name="DeliveryApptUpdateRequest">
            <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
            <wsdl:part element="tns:apptUpdate" name="apptUpdate"/>
        </wsdl:message>
        <wsdl:message name="DeliveryApptCancelCreateRequest">
            <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
            <wsdl:part element="tns:apptCancelCreate"
name="apptCancelCreate"/>
        </wsdl:message>
        <wsdl:message name="DeliveryApptCancelRequest">
            <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>

```

```

        <wsdl:part element="tns:apptCancel" name="apptCancel"/>
    </wsdl:message>
    <wsdl:message name="DeliveryApptShellRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
        <wsdl:part element="tns:apptShell" name="apptShell"/>
    </wsdl:message>
    <wsdl:message name="POAppointmentServicesFault">
        <wsdl:part element="tns:fault" name="fault"/>
    </wsdl:message>
    <wsdl:message name="DeliveryApptShellUpdateRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
        <wsdl:part element="tns:apptShellUpdate"
name="apptShellUpdate"/>
    </wsdl:message>
    <wsdl:message name="AppointmentServiceResponse">
        <wsdl:part element="tns:result" name="result"/>
    </wsdl:message>
    <wsdl:message name="PushSubscriptionResponse">
        <wsdl:part element="tns:result" name="result"/>
    </wsdl:message>
    <wsdl:message name="DeliveryApptShellCancelRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
        <wsdl:part element="tns:apptShellCancel"
name="apptShellCancel"/>
    </wsdl:message>
    <wsdl:message name="MessageResponseRetrievalRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
        <wsdl:part element="tns:messageResponseRetrieval"
name="messageResponseRetrieval"/>
    </wsdl:message>
    <wsdl:message name="MailingGroupQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo"/>
        <wsdl:part element="tns:mailingGroupQuery"
name="mailingGroupQuery"/>
    </wsdl:message>
    <wsdl:message name="PaymentMessageQueryRequest">

```

```

        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:paymentMessageQuery"
name="paymentMessageQuery" />
    </wsdl:message>
    <wsdl:message name="ReconciliationReportQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:reconciliationReportQuery"
name="reconciliationReportQuery" />
    </wsdl:message>
    <wsdl:message name="ContainerStatusQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:containerStatusQuery"
name="containerStatusQuery" />
    </wsdl:message>
    <wsdl:message name="PartnerApptQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:partnerApptQuery"
name="partnerApptQuery" />
    </wsdl:message>
    <wsdl:message name="CustomerSupplierAgreementQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:customerSupplierAgreementQuery"
name="customerSupplierAgreementQuery" />
    </wsdl:message>
    <wsdl:message name="FullServiceAddressCorrectionQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:fullServiceAddressCorrectionQuery"
name="fullServiceAddressCorrectionQuery" />
    </wsdl:message>
    <wsdl:message name="FullServiceContainerVisibilityQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:fullServiceContainerVisibilityQuery"
name="fullServiceContainerVisibilityQuery" />
    </wsdl:message>
    <wsdl:message name="FullServiceNixieDetailQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />

```

```

        <wsdl:part element="tns:fullServiceNixieDetailQuery"
name="fullServiceNixieDetailQuery" />
    </wsdl:message>
    <wsdl:message name="StartTheClockQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:startTheClockQuery"
name="startTheClockQuery" />
    </wsdl:message>
    <wsdl:message name="ByForConflictQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:byForConflictQuery"
name="byForConflictQuery" />
    </wsdl:message>
    <wsdl:message name="CastofCharactersCreateRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:castofCharactersCreate"
name="castofCharactersCreate" />
    </wsdl:message>
    <wsdl:message name="CastofCharactersUpdateRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:castofCharactersUpdate"
name="castofCharactersUpdate" />
    </wsdl:message>
    <wsdl:message name="CastofCharactersCancelRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:castofCharactersCancel"
name="castofCharactersCancel" />
    </wsdl:message>
    <wsdl:message name="DeliveryPointValidationQueryRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:deliveryPointValidationQuery"
name="deliveryPointValidationQuery" />
    </wsdl:message>
    <wsdl:message name="PushSubscriptionCreateRequest">
        <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
        <wsdl:part element="tns:pushSubscriptionCreate"
name="pushSubscriptionCreate" />

```

```

</wsdl:message>
  <wsdl:message name="PushSubscriptionUpdateRequest">
    <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
    <wsdl:part element="tns:pushSubscriptionUpdate"
name="pushSubscriptionUpdate" />
  </wsdl:message>
  <wsdl:message name="PushSubscriptionCancelRequest">
    <wsdl:part element="auth:AuthenticationInfo"
name="authInfo" />
    <wsdl:part element="tns:pushSubscriptionCancel"
name="pushSubscriptionCancel" />
  </wsdl:message>
  <wsdl:portType name="POAppointmentServicesPortType">
    <wsdl:operation name="QueryDeliveryAppt">
      <wsdl:input message="tns:DeliveryApptQueryRequest"
name="DeliveryApptQueryRequest" />
      <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptQueryResponse" />
      <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptQueryFault" />
    </wsdl:operation>
    <wsdl:operation name="RequestDeliveryAppt">
      <wsdl:input message="tns:DeliveryApptRequest"
name="DeliveryApptRequest" />
      <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptResponse" />
      <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptFault" />
    </wsdl:operation>
    <wsdl:operation name="UpdateDeliveryAppt">
      <wsdl:input message="tns:DeliveryApptUpdateRequest"
name="DeliveryApptUpdateRequest" />
      <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptUpdateResponse" />
      <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptUpdateFault" />
    </wsdl:operation>
    <wsdl:operation name="CancelDeliveryAppt">
      <wsdl:input message="tns:DeliveryApptCancelRequest"
name="DeliveryApptCancelRequest" />
      <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptCancelResponse" />

```

PostalOne! Mail.XML Technical Guide

```
        <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptCancelFault" />
    </wsdl:operation>
    <wsdl:operation name="GetDeliveryApptStatus">
        <wsdl:input message="tns:DeliveryApptStatusRequest"
name="DeliveryApptStatusRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptStatusResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptStatusFault" />
    </wsdl:operation>
    <wsdl:operation name="GetAllDeliveryApptCloseouts">
        <wsdl:input
message="tns:AllDeliveryApptCloseoutRequest"
name="AllDeliveryApptCloseoutRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse"
name="AllDeliveryApptCloseoutResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault"
name="AllDeliveryApptCloseoutFault" />
    </wsdl:operation>
    <wsdl:operation name="RetrieveAppointmentResponses">
        <wsdl:input message="tns:ApptResponseRetrieval"
name="ApptResponseRetrievalRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse"
name="ApptResponseRetrievalResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault"
name="ApptResponseRetrievalFault" />
    </wsdl:operation>
    <wsdl:operation name="DeliveryApptContentCreate">
        <wsdl:input
message="tns:DeliveryApptContentCreateRequest"
name="DeliveryApptContentCreateRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptContentCreateResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptContentCreateFault" />
    </wsdl:operation>
    <wsdl:operation name="DeliveryApptContentUpdate">
        <wsdl:input
message="tns:DeliveryApptContentUpdateRequest"
name="DeliveryApptContentUpdateRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptContentUpdateResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptContentUpdateFault" />
    </wsdl:operation>
```

```

        <wsdl:operation name="DeliveryApptContentCancel">
            <wsdl:input
message="tns:DeliveryApptContentCancelRequest "
name="DeliveryApptContentCancelRequest" />
            <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptContentCancelResponse" />
            <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptContentCancelFault" />
        </wsdl:operation>
        <wsdl:operation name="DeliveryContentQuery">
            <wsdl:input message="tns:DeliveryContentQueryRequest"
name="DeliveryContentQueryRequest" />
            <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryContentQueryResponse" />
            <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryContentQueryFault" />
        </wsdl:operation>
        <wsdl:operation name="DeliveryApptShell">
            <wsdl:input message="tns:DeliveryApptShellRequest"
name="DeliveryApptShellRequest" />
            <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptShellResponse" />
            <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptShellFault" />
        </wsdl:operation>
        <wsdl:operation name="DeliveryApptShellUpdate">
            <wsdl:input
message="tns:DeliveryApptShellUpdateRequest "
name="DeliveryApptShellUpdateRequest" />
            <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptShellUpdateResponse" />
            <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptShellUpdateFault" />
        </wsdl:operation>
        <wsdl:operation name="DeliveryApptShellCancel">
            <wsdl:input
message="tns:DeliveryApptShellCancelRequest "
name="DeliveryApptShellCancelRequest" />
            <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryApptShellCancelResponse" />
            <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryApptShellCancelFault" />
        </wsdl:operation>
        <wsdl:operation name="DeliveryApptCancelCreate">

```

```

        <wsdl:input
message="tns:DeliveryApptCancelCreateRequest "
name="DeliveryApptCancelCreateRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse "
name="DeliveryApptCancelCreateResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault "
name="DeliveryApptCancelCreateFault" />
    </wsdl:operation>
    <wsdl:operation name="RecurringApptQuery">
        <wsdl:input message="tns:RecurringApptQueryRequest "
name="RecurringApptQueryRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse "
name="RecurringApptQueryResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault "
name="RecurringApptQueryFault" />
    </wsdl:operation>
    <wsdl:operation name="MessageResponseRetrieval">
        <wsdl:input
message="tns:MessageResponseRetrievalRequest "
name="MessageResponseRetrievalRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse "
name="MessageResponseRetrievalResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault "
name="MessageResponseRetrievalFault" />
    </wsdl:operation>
    <wsdl:operation name="MailingGroupQuery">
        <wsdl:input message="tns:MailingGroupQueryRequest "
name="MailingGroupQueryRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse "
name="MailingGroupQueryResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault "
name="MailingGroupQueryFault" />
    </wsdl:operation>
    <wsdl:operation name="PaymentMessageQuery">
        <wsdl:input message="tns:PaymentMessageQueryRequest "
name="PaymentMessageQueryRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse "
name="PaymentMessageQueryResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault "
name="PaymentMessageQueryFault" />
    </wsdl:operation>
    <wsdl:operation name="ReconciliationReportQuery">
        <wsdl:input
message="tns:ReconciliationReportQueryRequest "
name="ReconciliationReportQueryRequest" />

```

PostalOne! Mail.XML Technical Guide

```
<wsdl:output message="tns:AppointmentServiceResponse"
name="ReconciliationReportQueryResponse" />
    <wsdl:fault message="tns:POAppointmentServicesFault"
name="ReconciliationReportQueryFault" />
</wsdl:operation>
    <wsdl:operation name="ContainerStatusQuery">
        <wsdl:input message="tns:ContainerStatusQueryRequest"
name="ContainerStatusQueryRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse"
name="ContainerStatusQueryResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault"
name="ContainerStatusQueryFault" />
    </wsdl:operation>
    <wsdl:operation name="PartnerApptQuery">
        <wsdl:input message="tns:PartnerApptQueryRequest"
name="PartnerApptQueryRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse"
name="PartnerApptQueryResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault"
name="PartnerApptQueryFault" />
    </wsdl:operation>
    <wsdl:operation name="CustomerSupplierAgreementQuery">
        <wsdl:input
message="tns:CustomerSupplierAgreementQueryRequest "
name="CustomerSupplierAgreementQueryRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse"
name="CustomerSupplierAgreementQueryResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault"
name="CustomerSupplierAgreementQueryFault" />
    </wsdl:operation>
    <wsdl:operation name="FullServiceAddressCorrectionQuery">
        <wsdl:input
message="tns:FullServiceAddressCorrectionQueryRequest "
name="FullServiceAddressCorrectionQueryRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse"
name="FullServiceAddressCorrectionQueryResponse" />
        <wsdl:fault message="tns:POAppointmentServicesFault"
name="FullServiceAddressCorrectionQueryFault" />
    </wsdl:operation>
    <wsdl:operation name="FullServiceContainerVisibilityQuery">
        <wsdl:input
message="tns:FullServiceContainerVisibilityQueryRequest "
name="FullServiceContainerVisibilityQueryRequest" />
        <wsdl:output message="tns:AppointmentServiceResponse"
name="FullServiceContainerVisibilityQueryResponse" />
```

PostalOne! Mail.XML Technical Guide

```
<wsdl:fault message="tns:POAppointmentServicesFault"
name="FullServiceContainerVisibilityQueryFault" />
</wsdl:operation>
<wsdl:operation name="FullServiceNixieDetailQuery">
  <wsdl:input
message="tns:FullServiceNixieDetailQueryRequest"
name="FullServiceNixieDetailQueryRequest" />
  <wsdl:output message="tns:AppointmentServiceResponse"
name="FullServiceNixieDetailQueryResponse" />
  <wsdl:fault message="tns:POAppointmentServicesFault"
name="FullServiceNixieDetailQueryFault" />
</wsdl:operation>
<wsdl:operation name="StartTheClockQuery">
  <wsdl:input message="tns:StartTheClockQueryRequest"
name="StartTheClockQueryRequest" />
  <wsdl:output message="tns:AppointmentServiceResponse"
name="StartTheClockQueryResponse" />
  <wsdl:fault message="tns:POAppointmentServicesFault"
name="StartTheClockQueryFault" />
</wsdl:operation>
<wsdl:operation name="ByForConflictQuery">
  <wsdl:input message="tns:ByForConflictQueryRequest"
name="ByForConflictQueryRequest" />
  <wsdl:output message="tns:AppointmentServiceResponse"
name="ByForConflictQueryResponse" />
  <wsdl:fault message="tns:POAppointmentServicesFault"
name="ByForConflictQueryFault" />
</wsdl:operation>
<wsdl:operation name="CastofCharactersCreate">
  <wsdl:input message="tns:CastofCharactersCreateRequest"
name="CastofCharactersCreateRequest" />
  <wsdl:output message="tns:AppointmentServiceResponse"
name="CastofCharactersCreateResponse" />
  <wsdl:fault message="tns:POAppointmentServicesFault"
name="CastofCharactersCreateFault" />
</wsdl:operation>
<wsdl:operation name="CastofCharactersUpdate">
  <wsdl:input message="tns:CastofCharactersUpdateRequest"
name="CastofCharactersUpdateRequest" />
  <wsdl:output message="tns:AppointmentServiceResponse"
name="CastofCharactersUpdateResponse" />
  <wsdl:fault message="tns:POAppointmentServicesFault"
name="CastofCharactersUpdateFault" />
</wsdl:operation>
<wsdl:operation name="CastofCharactersCancel">
```

PostalOne! Mail.XML Technical Guide

```
<wsdl:input message="tns:CastofCharactersCancelRequest"
name="CastofCharactersCancelRequest" />
    <wsdl:output message="tns:AppointmentServiceResponse"
name="CastofCharactersCancelResponse" />
    <wsdl:fault message="tns:POAppointmentServicesFault"
name="CastofCharactersCancelFault" />
</wsdl:operation>
<wsdl:operation name="DeliveryPointValidationQuery">
    <wsdl:input
message="tns:DeliveryPointValidationQueryRequest"
name="DeliveryPointValidationQueryRequest" />
    <wsdl:output message="tns:AppointmentServiceResponse"
name="DeliveryPointValidationQueryResponse" />
    <wsdl:fault message="tns:POAppointmentServicesFault"
name="DeliveryPointValidationQueryFault" />
</wsdl:operation>
<wsdl:operation name="PushSubscriptionCreate">
    <wsdl:input message="tns:PushSubscriptionCreateRequest"
name="PushSubscriptionCreateRequest" />
    <wsdl:output message="tns:PushSubscriptionResponse"
name="PushSubscriptionCreateResponse" />
    <wsdl:fault message="tns:POAppointmentServicesFault"
name="PushSubscriptionCreateFault" />
</wsdl:operation>
<wsdl:operation name="PushSubscriptionUpdate">
    <wsdl:input message="tns:PushSubscriptionUpdateRequest"
name="PushSubscriptionUpdateRequest" />
    <wsdl:output message="tns:PushSubscriptionResponse"
name="PushSubscriptionUpdateResponse" />
    <wsdl:fault message="tns:POAppointmentServicesFault"
name="PushSubscriptionUpdateFault" />
</wsdl:operation>
<wsdl:operation name="PushSubscriptionCancel">
    <wsdl:input message="tns:PushSubscriptionCancelRequest"
name="PushSubscriptionCancelRequest" />
    <wsdl:output message="tns:PushSubscriptionResponse"
name="PushSubscriptionCancelResponse" />
    <wsdl:fault message="tns:POAppointmentServicesFault"
name="PushSubscriptionCancelFault" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="POAppointmentServicesSoapBinding"
type="tns:POAppointmentServicesPortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
```

PostalOne! Mail.XML Technical Guide

```
<wsdl:operation name="QueryDeliveryAppt">
  <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/queryDeliveryAppt"/>
  <wsdl:input name="DeliveryApptQueryRequest">
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="DeliveryApptQueryResponse">
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="DeliveryApptQueryFault">
    <soap:fault name="DeliveryApptQueryFault"
use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="RequestDeliveryAppt">
  <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/requestDeliveryAppt"/>
  <wsdl:input name="DeliveryApptRequest">
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="DeliveryApptResponse">
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="DeliveryApptFault">
    <soap:fault name="DeliveryApptFault"
use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="UpdateDeliveryAppt">
  <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/updateDeliveryAppt"/>
  <wsdl:input name="DeliveryApptUpdateRequest">
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="DeliveryApptUpdateResponse">
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="DeliveryApptUpdateFault">
```

```

                <soap:fault name="DeliveryApptUpdateFault"
use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="CancelDeliveryAppt">
            <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/cancelDeliveryAppt"/>
            <wsdl:input name="DeliveryApptCancelRequest">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="DeliveryApptCancelResponse">
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="DeliveryApptCancelFault">
                <soap:fault name="DeliveryApptCancelFault"
use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="GetDeliveryApptStatus">
            <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/getDeliveryApptStatus"/>
            <wsdl:input name="DeliveryApptStatusRequest">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="DeliveryApptStatusResponse">
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="DeliveryApptStatusFault">
                <soap:fault name="DeliveryApptStatusFault"
use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="GetAllDeliveryApptCloseouts">
            <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/getAllDeliveryApptCloseouts"/>
            <wsdl:input name="AllDeliveryApptCloseoutRequest">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="AllDeliveryApptCloseoutResponse">

```

```

        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="AllDeliveryApptCloseoutFault">
        <soap:fault name="AllDeliveryApptCloseoutFault"
use="literal"/>
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="RetrieveAppointmentResponses">
    <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/retrieveAppointmentResponses"/>
    <wsdl:input name="ApptResponseRetrievalRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="ApptResponseRetrievalResponse">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ApptResponseRetrievalFault">
        <soap:fault name="ApptResponseRetrievalFault"
use="literal"/>
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="DeliveryApptContentCreate">
    <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/DeliveryApptContentCreate"/>
    <wsdl:input name="DeliveryApptContentCreateRequest">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="DeliveryApptContentCreateResponse">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="DeliveryApptContentCreateFault">
        <soap:fault name="DeliveryApptContentCreateFault"
use="literal"/>
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="DeliveryApptContentUpdate">
    <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/DeliveryApptContentUpdate"/>
    <wsdl:input name="DeliveryApptContentUpdateRequest">

```

PostalOne! Mail.XML Technical Guide

```
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output name="DeliveryApptContentUpdateResponse">
        <soap:body use="literal" />
    </wsdl:output>
    <wsdl:fault name="DeliveryApptContentUpdateFault">
        <soap:fault name="DeliveryApptContentUpdateFault"
use="literal" />
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="DeliveryApptContentCancel">
    <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/DeliveryApptContentCancel" />
    <wsdl:input name="DeliveryApptContentCancelRequest">
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output name="DeliveryApptContentCancelResponse">
        <soap:body use="literal" />
    </wsdl:output>
    <wsdl:fault name="DeliveryApptContentCancelFault">
        <soap:fault name="DeliveryApptContentCancelFault"
use="literal" />
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="DeliveryContentQuery">
    <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/DeliveryContentQuery" />
    <wsdl:input name="DeliveryContentQueryRequest">
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output name="DeliveryContentQueryResponse">
        <soap:body use="literal" />
    </wsdl:output>
    <wsdl:fault name="DeliveryContentQueryFault">
        <soap:fault name="DeliveryContentQueryFault"
use="literal" />
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="DeliveryApptShell">
```

PostalOne! Mail.XML Technical Guide

```

        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/DeliveryApptShell"/>
        <wsdl:input name="DeliveryApptShellRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="DeliveryApptShellResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="DeliveryApptShellFault">
            <soap:fault name="DeliveryApptShellFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="DeliveryApptShellUpdate">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/DeliveryApptShellUpdate"/>
        <wsdl:input name="DeliveryApptShellUpdateRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="DeliveryApptShellUpdateResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="DeliveryApptShellUpdateFault">
            <soap:fault name="DeliveryApptShellUpdateFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="DeliveryApptShellCancel">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/DeliveryApptShellCancel"/>
        <wsdl:input name="DeliveryApptShellCancelRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="DeliveryApptShellCancelResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="DeliveryApptShellCancelFault">
            <soap:fault name="DeliveryApptShellCancelFault"
use="literal"/>
    </wsdl:operation>

```

```

        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="DeliveryApptCancelCreate">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/DeliveryApptCancelCreate"/>
        <wsdl:input name="DeliveryApptCancelCreateRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="DeliveryApptCancelCreateResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="DeliveryApptCancelCreateFault">
            <soap:fault name="DeliveryApptCancelCreateFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="RecurringApptQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/RecurringApptQuery"/>
        <wsdl:input name="RecurringApptQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="RecurringApptQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="RecurringApptQueryFault">
            <soap:fault name="RecurringApptQueryFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="MessageResponseRetrieval">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/MessageResponseRetrieval"/>
        <wsdl:input name="MessageResponseRetrievalRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="MessageResponseRetrievalResponse">
            <soap:body use="literal"/>
        </wsdl:output>

```

```

        <wsdl:fault name="MessageResponseRetrievalFault">
            <soap:fault name="MessageResponseRetrievalFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="MailingGroupQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/MailingGroupQuery"/>
        <wsdl:input name="MailingGroupQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="MailingGroupQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="MailingGroupQueryFault">
            <soap:fault name="MailingGroupQueryFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="PaymentMessageQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/PaymentMessageQuery"/>
        <wsdl:input name="PaymentMessageQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="PaymentMessageQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="PaymentMessageQueryFault">
            <soap:fault name="PaymentMessageQueryFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="ReconciliationReportQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/ReconciliationReportQuery"/>
        <wsdl:input name="ReconciliationReportQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
    </wsdl:operation>

```

```

        <wsdl:output name="ReconciliationReportQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="ReconciliationReportQueryFault">
            <soap:fault name="ReconciliationReportQueryFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="ContainerStatusQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/ContainerStatusQuery"/>
        <wsdl:input name="ContainerStatusQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="ContainerStatusQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="ContainerStatusQueryFault">
            <soap:fault name="ContainerStatusQueryFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="PartnerApptQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/PartnerApptQuery"/>
        <wsdl:input name="PartnerApptQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="PartnerApptQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="PartnerApptQueryFault">
            <soap:fault name="PartnerApptQueryFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="CustomerSupplierAgreementQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/AppointmentService
/CustomerSupplierAgreementQuery"/>

```

PostalOne! Mail.XML Technical Guide

```
        <wsdl:input
name="CustomerSupplierAgreementQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output
name="CustomerSupplierAgreementQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="CustomerSupplierAgreementQueryFault">
            <soap:fault
name="CustomerSupplierAgreementQueryFault" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="FullServiceAddressCorrectionQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/FullServiceAddress
CorrectionQuery"/>
        <wsdl:input
name="FullServiceAddressCorrectionQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output
name="FullServiceAddressCorrectionQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault
name="FullServiceAddressCorrectionQueryFault">
            <soap:fault
name="FullServiceAddressCorrectionQueryFault" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="FullServiceContainerVisibilityQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/FullServiceContain
erVisibilityQuery"/>
        <wsdl:input
name="FullServiceContainerVisibilityQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output
name="FullServiceContainerVisibilityQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
```

PostalOne! Mail.XML Technical Guide

```
        <wsdl:fault
name="FullServiceContainerVisibilityQueryFault">
            <soap:fault
name="FullServiceContainerVisibilityQueryFault" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="FullServiceNixieDetailQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/FullServiceNixieDe
tailQuery"/>
        <wsdl:input name="FullServiceNixieDetailQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output
name="FullServiceNixieDetailQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="FullServiceNixieDetailQueryFault">
            <soap:fault
name="FullServiceNixieDetailQueryFault" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="StartTheClockQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/StartTheClockQuery
"/>
        <wsdl:input name="StartTheClockQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="StartTheClockQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="StartTheClockQueryFault">
            <soap:fault name="StartTheClockQueryFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="ByForConflictQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/ByForConflictQuery
"/>
        <wsdl:input name="ByForConflictQueryRequest">
            <soap:body use="literal"/>
```

```

        </wsdl:input>
        <wsdl:output name="ByForConflictQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="ByForConflictQueryFault">
            <soap:fault name="ByForConflictQueryFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="CastofCharactersCreate">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/CastofCharactersCr
eate"/>
        <wsdl:input name="CastofCharactersCreateRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="CastofCharactersCreateResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="CastofCharactersCreateFault">
            <soap:fault name="CastofCharactersCreateFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="CastofCharactersUpdate">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/CastofCharactersUp
date"/>
        <wsdl:input name="CastofCharactersUpdateRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="CastofCharactersUpdateResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="CastofCharactersUpdateFault">
            <soap:fault name="CastofCharactersUpdateFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="CastofCharactersCancel">

```

PostalOne! Mail.XML Technical Guide

```

        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/CastofCharactersCa
ncel"/>
        <wsdl:input name="CastofCharactersCancelRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="CastofCharactersCancelResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="CastofCharactersCancelFault">
            <soap:fault name="CastofCharactersCancelFault"
use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="DeliveryPointValidationQuery">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/DeliveryPointValid
ationQuery"/>
        <wsdl:input name="DeliveryPointValidationQueryRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output
name="DeliveryPointValidationQueryResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="DeliveryPointValidationQueryFault">
            <soap:fault
name="DeliveryPointValidationQueryFault" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="PushSubscriptionCreate">
        <soap:operation
soapAction="https://www.uspspostalone.com/ops/services/PushSubscriptionCr
eate"/>
        <wsdl:input name="PushSubscriptionCreateRequest">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="PushSubscriptionCreateResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="PushSubscriptionCreateFault">

```



```

</wsdl:service>
<!--Push Subscriptions Messages (Data Distribution) -->
<!--Push Subscriptions Messages (Data Distribution) -->
</wsdl:definitions>

```

+++++

PUSH User Profile

+++++

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:pdd="http://www.usps.com/postalone/services/PushUserProfileSchema"
targetNamespace="http://www.usps.com/postalone/services/PushUserProfileSc
hema" version="1.0.0">
  <xs:annotation>
    <xs:documentation xml:lang="en">PostalOne! business entity
subscription</xs:documentation>
  </xs:annotation>
  <xs:element name="CreateProfileRequest"
type="pdd:createProfileRequestType" />
  <xs:element name="UpdateProfileRequest"
type="pdd:updateProfileRequestType" />
  <xs:element name="CancelProfileRequest"
type="pdd:cancelProfileRequestType" />
  <xs:element name="ProfileResponse" type="pdd:profileResponseType" />
  <!-- New subscription request -->
  <xs:complexType name="createProfileRequestType">
    <xs:sequence>
      <xs:element name="BusinessEntity"
type="pdd:businessEntityType" />
      <xs:element name="MessageType" type="pdd:messageType" />
      <xs:element name="PushType" type="pdd:pushType" />
      <xs:element name="ExternalSystemUserId"
type="pdd:usernameType" />
      <xs:element name="ExternalSystemPassword"
type="pdd:passwordType" />
      <xs:element name="ExternalSystemConfirmPassword"
type="pdd:passwordType" />
      <xs:element name="ExternalSystemURL" type="xs:string" />
      <xs:element name="PushStartDate" type="pdd:dateType" />
      <xs:element name="PushFrequency"
type="pdd:pushFrequencyType" />
    </xs:sequence>
  </xs:complexType>
  <!-- Edit an existing subscription request -->

```

```

<xs:complexType name="updateProfileRequestType">
  <xs:sequence>
    <xs:element name="BusinessEntity"
type="pdd:businessEntityType"/>
    <xs:element name="MessageType" type="pdd:messageType"/>
    <xs:element name="PushType" type="pdd:pushType"/>
    <xs:element name="ExternalSystemUserId"
type="pdd:usernameType"/>
    <xs:element name="ExternalSystemPassword"
type="pdd:passwordType"/>
    <xs:element name="ExternalSystemConfirmPassword"
type="pdd:passwordType"/>
    <xs:element name="ExternalSystemURL" type="xs:string"/>
    <xs:element name="PushStartDate" type="pdd:dateType"/>
    <xs:element name="PushFrequency"
type="pdd:pushFrequencyType"/>
  </xs:sequence>
</xs:complexType>
<!-- Unsubscribing a subscription request -->
<xs:complexType name="cancelProfileRequestType">
  <xs:sequence>
    <xs:element name="BusinessEntity"
type="pdd:businessEntityType"/>
    <xs:element name="MessageType" type="pdd:messageType"/>
  </xs:sequence>
</xs:complexType>
<!-- Successful or Error responses -->
<xs:complexType name="profileResponseType">
  <xs:sequence>
    <xs:element minOccurs="0" name="RequestString"
type="xs:string"/>
    <xs:element name="Status"
type="pdd:responseStatusType"/>
    <xs:element minOccurs="0" name="Error"
type="pdd:errorType"/>
  </xs:sequence>
</xs:complexType>
<!-- The sender needs to provide a key/id which can identify a
business entity uniquely -->
<xs:complexType name="businessEntityType">
  <xs:sequence>
    <xs:element name="LocationId" type="xs:string"/>

```

```

        </xs:sequence>
</xs:complexType>
<xs:simpleType name="messageType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Address Correction"/>
        <xs:enumeration value="Container Visibility"/>
        <xs:enumeration value="Nixie Detail"/>
        <xs:enumeration value="Start the Clock"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="pushType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Delivery"/>
        <xs:enumeration value="Notification"/>
    </xs:restriction>
</xs:simpleType>
<!-- User id of the external system getting the data by push -->
<xs:simpleType name="usernameType">
    <xs:restriction base="xs:string">
        <xs:minLength value="2"/>
        <xs:maxLength value="30"/>
    </xs:restriction>
</xs:simpleType>
<!-- Password of the external system getting the data by push -->
<xs:simpleType name="passwordType">
    <xs:restriction base="xs:string">
        <xs:minLength value="2"/>
        <xs:maxLength value="40"/>
    </xs:restriction>
</xs:simpleType>
<!-- The date should be formatted as MM/DD/YYYY, i.e. 09/23/2007 --
>
<xs:simpleType name="dateType">
    <xs:restriction base="xs:string">
        <xs:pattern value="[0-1][0-9][/][0-3][0-9][/][0-2][0-
2][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="pushFrequencyType">

```

```

        <xs:restriction base="xs:string">
            <xs:enumeration value="Every Six Hours"/>
            <xs:enumeration value="Every Twelve Hours"/>
            <xs:enumeration value="Every Twenty Four Hours"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="errorType">
        <xs:sequence>
            <xs:element name="Code" type="xs:string"/>
            <xs:element name="Details" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="responseStatusType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="success"/>
            <xs:enumeration value="error"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>

```

+++++

User Authentication Info XSD

+++++

```

<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:auth="http://www.usps.com/postalone/services/UserAuthenticationSche
ma"
targetNamespace="http://www.usps.com/postalone/services/UserAuthenticatio
nSchema" version="0.0.3">

    <xsd:annotation>
        <xsd:documentation xml:lang="en">PostalOne! User
Authentication Request</xsd:documentation>
    </xsd:annotation>

    <xsd:element name="AuthenticationInfo"
type="auth:AuthenticationInfoType"/>

    <xsd:complexType name="AuthenticationInfoType">
        <xsd:sequence>
            <xsd:element name="UserId" type="xsd:string"/>
            <xsd:element name="UserPassword" type="xsd:string"/>

```

```

        </xsd:sequence>
    </xsd:complexType>

</xsd:schema>

```

WSDL - Required for the PUSH Model to Receive Data from the *PostalOne!* System

```

<?xml version="1.0" encoding="UTF-8"?>
<!--      Date          Comments          -->
<!--      11/17/2008    POCustomerMailXMLServices WSDL  Initial version 1.0
-->

<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.usps.com/postalone/services/POCustomerMailXMLServices"
xmlns:auth="http://www.usps.com/postalone/services/UserAuthenticationSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://www.usps.com/postalone/services/POCustomerMailXMLServices"
name="POCustomerMailXMLServices"
targetNamespace="http://www.usps.com/postalone/services/POCustomerMailXMLServices">

    <wsdl:documentation
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Version
1.0.0</wsdl:documentation>

    <wsdl:types>

        <xsd:schema
targetNamespace="http://www.usps.com/postalone/services/POCustomerMailXMLServices"
xmlns="http://www.usps.com/postalone/services/POCustomerMailXMLServices"
xmlns:auth="http://www.usps.com/postalone/services/UserAuthenticationSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.usps.com/postalone/services/POCustomerMailXMLServices"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

            <xsd:import
namespace="http://www.usps.com/postalone/services/UserAuthenticationSchema"
schemaLocation="UserAuthenticationInfo.xsd"/>

            <xsd:element name="fullServiceAddressCorrectionData"
type="xsd:string"/>

            <xsd:element name="fullServiceContainerVisibilityData"
type="xsd:string"/>

            <xsd:element name="fullServiceNixieDetailData"
type="xsd:string"/>

            <xsd:element name="DeliveryPointVerificationData"
type="xsd:string"/>

            <xsd:element name="StartTheClockData"
type="xsd:string"/>

            <xsd:element name="acknowledgement" type="xsd:string"/>

```

```

        <xsd:element name="fault" type="xsd:string"/>
    </xsd:schema>
</wsdl:types>
<wsdl:message name="AddressCorrectionData">
    <wsdl:part name="authInfo"
element="auth:AuthenticationInfo"/>
    <wsdl:part name="addressCorrectionData"
element="tns:fullServiceAddressCorrectionData"/>
</wsdl:message>
<wsdl:message name="ContainerVisibilityData">
    <wsdl:part name="authInfo"
element="auth:AuthenticationInfo"/>
    <wsdl:part name="containerVisibilityData"
element="tns:fullServiceContainerVisibilityData"/>
</wsdl:message>
<wsdl:message name="NixieDetailData">
    <wsdl:part name="authInfo"
element="auth:AuthenticationInfo"/>
    <wsdl:part name="nixieDetailData"
element="tns:fullServiceNixieDetailData"/>
</wsdl:message>
<wsdl:message name="DeliveryPointVerificationData">
    <wsdl:part name="authInfo"
element="auth:AuthenticationInfo"/>
    <wsdl:part name="deliveryPointVerificationData"
element="tns:DeliveryPointVerificationData"/>
</wsdl:message>
<wsdl:message name="StartTheClockData">
    <wsdl:part name="authInfo"
element="auth:AuthenticationInfo"/>
    <wsdl:part name="startTheClockData"
element="tns:StartTheClockData"/>
</wsdl:message>
<wsdl:message name="Acknowledgement">
    <wsdl:part name="acknowledgement"
element="tns:acknowledgement"/>
</wsdl:message>
<wsdl:message name="POCustomerMailXMLServicesFault">
    <wsdl:part name="fault" element="tns:fault"/>
</wsdl:message>
<wsdl:portType name="POCustomerMailXMLServicesPortType">
    <wsdl:operation name="PushAddressCorrectionData">

```

PostalOne! Mail.XML Technical Guide

```
        <wsdl:input name="AddressCorrectionData"
message="tns:AddressCorrectionData" />
        <wsdl:output name="AddressCorrectionAcknowledgement"
message="tns:Acknowledgement" />
        <wsdl:fault name="PushAddressCorrectionDataFault"
message="tns:POCustomerMailXMLServicesFault" />
    </wsdl:operation>
    <wsdl:operation name="PushContainerVisibilityData">
        <wsdl:input name="ContainerVisibilityData"
message="tns:ContainerVisibilityData" />
        <wsdl:output name="ContainerVisibilityAcknowledgement"
message="tns:Acknowledgement" />
        <wsdl:fault name="PushContainerVisibilityDataFault"
message="tns:POCustomerMailXMLServicesFault" />
    </wsdl:operation>
    <wsdl:operation name="PushNixieDetailData">
        <wsdl:input name="NixieDetailData"
message="tns:NixieDetailData" />
        <wsdl:output name="NixieDetailAcknowledgement"
message="tns:Acknowledgement" />
        <wsdl:fault name="PushNixieDetailDataFault"
message="tns:POCustomerMailXMLServicesFault" />
    </wsdl:operation>
    <wsdl:operation name="PushDeliveryPointVerificationData">
        <wsdl:input name="DeliveryPointVerificationData"
message="tns:DeliveryPointVerificationData" />
        <wsdl:output
name="DeliveryPointVerificationAcknowledgement"
message="tns:Acknowledgement" />
        <wsdl:fault
name="PushDeliveryPointVerificationDataFault"
message="tns:POCustomerMailXMLServicesFault" />
    </wsdl:operation>
    <wsdl:operation name="PushStartTheClockData">
        <wsdl:input name="StartTheClockData"
message="tns:StartTheClockData" />
        <wsdl:output name="StartTheClockAcknowledgement"
message="tns:Acknowledgement" />
        <wsdl:fault name="PushStartTheClockDataFault"
message="tns:POCustomerMailXMLServicesFault" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="POCustomerMailXMLServicesSoapBinding"
type="tns:POCustomerMailXMLServicesPortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
```

PostalOne! Mail.XML Technical Guide

```
<wsdl:operation name="PushAddressCorrectionData">
    <soap:operation
soapAction="https://www.uspspostalone.com/POCustomerMailXMLServices/pushA
ddressCorrectionData"/>
    <wsdl:input name="AddressCorrectionData">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="AddressCorrectionAcknowledgement">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="PushAddressCorrectionDataFault">
        <soap:fault name="PushAddressCorrectionDataFault"
use="literal"/>
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="PushContainerVisibilityData">
    <soap:operation
soapAction="https://www.uspspostalone.com/POCustomerMailXMLServices/pushC
ontainerVisibilityData"/>
    <wsdl:input name="ContainerVisibilityData">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="ContainerVisibilityAcknowledgement">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="PushContainerVisibilityDataFault">
        <soap:fault
name="PushContainerVisibilityDataFault" use="literal"/>
    </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="PushNixieDetailData">
    <soap:operation
soapAction="https://www.uspspostalone.com/POCustomerMailXMLServices/pushN
ixieDetailData"/>
    <wsdl:input name="NixieDetailData">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="NixieDetailAcknowledgement">
        <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="PushNixieDetailDataFault">
```



```
</wsdl:service>  
</wsdl:definitions>
```

+++++

User Authentication XSD -- Required to allow the *Postalone!* System to Invoke Customer's Web Services

+++++

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsd:schema  
  xmlns:auth="http://www.usps.com/postalone/services/UserAuthenticationSche  
ma" xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  targetNamespace="http://www.usps.com/postalone/services/UserAuthenticatio  
nSchema" version="0.0.3">  
  <xsd:annotation>  
    <xsd:documentation xml:lang="en">PostalOne! User  
Authentication Request</xsd:documentation>  
  </xsd:annotation>  
  <xsd:element name="AuthenticationInfo"  
type="auth:AuthenticationInfoType"/>  
  <xsd:complexType name="AuthenticationInfoType">  
    <xsd:sequence>  
      <xsd:element name="UserId" type="xsd:string"/>  
      <xsd:element name="UserPassword" type="xsd:string"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:schema>
```

+++++

Appendix B – Message Choreography for Data Distribution

No	Message Name	Description	Customer	USPS <i>PostalOne!</i>
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 dropshipped XML block, and BMEU entered mail data in the BMEU XML block.		From
3	FullServiceContainerVisibilityQueryRequest	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	FullServiceContainerVisibilityQueryResponse	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	FullServiceAddressCorrectionQueryRequest	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	FullServiceAddressCorrectionQueryResponse	The USPS, in response to the requests, sends all the address correction data pertinent to the request.		From
7	FullServiceNixieDetailQueryRequest	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	FullServiceNixieDetailResponse	The USPS, in response to the requests, sends all the Nixie detail data by MID and Job information back to the customer.		From
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

No	Message Name	Description	Customer	USPS <i>PostalOne!</i>
12	FullServiceContainerVisibilityNotification	The USPS sends this notification to the customer and provides the Job information for which the ContainerVisibility data is ready		From
13	FullServiceContainerVisibilityDelivery	The USPS sends this detail data delivery to the customer with all the data available in the FullServiceContainerVisibilityQueryResponse message		From
14	FullServiceAddressCorrectionNotification	The USPS sends this notification to the customer and provides the Job information for which the AddressCorrection data is ready		From
15	FullServiceAddressCorrectionDelivery	The USPS sends this detail data delivery to the customer with all the data available in the FullServiceAddressCorrectionQueryResponse message		From
16	FullServiceNixieDetailNotification	The USPS sends this notification to the customer and provides the Job information for which the NixieDetail data is ready		From
17	FullServiceNixieDetailDelivery	The USPS sends this detail data delivery to the customer with all the data available in the FullServiceNixieDetailQueryResponse message		From
18	NotificationResponse	This message is used in response to all messages that end in the word 'Notification'	From	
19	DeliveryResponse	This message is used in response to all messages that end in the word 'Delivery'	From	

Appendix C - eDocumentation Form Values

The appendix lists the values associated with the following Mail.XML 6.0 messages: MailingGroupQueryRequest, MailingGroupQueryResponse, PaymentMessageQueryRequest, PaymentMessageQueryResponse, ReconciliationReportQueryRequest, ReconciliationReportQueryResponse, ContainerStatusQueryRequest, and ContainerStatusQueryResponse.

eDoc Form Values

Container Status Type

Entry Facility Type Values

Value	Description
Accepted	Accepted for Induction, passed validation
OnHold	Waiting for Induction validation to complete
Rejected	Didn't satisfy the induction validation rules

Entry Facility Type

Entry Facility Type Values

Value	Description
O	Origin
B	BMC
S	SCF
D	DDU

Form Type

Form Type Values

Value	Description	Postage Statement
PM	Priority Mail	3600
FC	First-Class Mail	3600
SM	Standard Mail	3602
BP	Bound Printed Matter	3605
PP	Package Services	3605
PE	Periodicals	3541
ML	Media/Library Mail	3605

Full-Service Indicator

Full-Service Indicator Values

Value	Description
Blank/Null/Nil	Null
F	Full-Service
B	Basic
P	PostNet

Value	Description
O	Other

Rate Category

Rate Category Values

Value	Description
Blank/Null/Nil	Null
A	Saturation ECR
AF	Standard Mail Saturation ECR Letter Paid As Flat
B	High Density ECR
BF	Standard Mail High Density ECR Letter Paid As Flat
D	Basic ECR (Carrier Route)
DF	Standard Mail Basic ECR Letter Paid As Flat
E	5-Digit Automation (5-Digit Barcode)
FB	Firm Bundle Outside County
G	5-Digit Non Barcode(Auto)
GF	5-Digit Non Barcode (Auto) Letter Paid As Flat
H	3-Digit Automation (3-Digit Barcode)
K	3-Digit Non Barcode(Auto)
KF	3-Digit Non Barcode (Auto) Letter Paid As Flat
L	Basic Automation (Basic Barcode)
L1	AADC Barcode
L2	Mixed AADC Barcode
L3	ADC Barcode
L4	Mixed ADC Barcode
L5	ADC Non-Barcode
L6	Mixed ADC Non-Barcode
L7	AADC Non-Barcode
L8	Mixed AADC Non-Barcode
N	Basic Nonautomation (Basic Presorted)
NG	5-Digit Not Flat Machinable
NK	3-Digit Not Flat Machinable
N5	ADC/BMC Not Flat Machinable
N6	Mixed ADC/BMC Not Flat Machinable
PI	5-Digit Parcel 0.2063 pounds or less
PM	3-Digit Parcel 0.2063 pounds or less
P7	ADC Parcel 0.2063 pounds or less
P8	Mixed ADC Parcel 0.2063 pounds or less
PE	5-Digit Machinable Parcel over 0.2063 pounds
PQ	BMC Machinable Parcel over 0.2063 pounds
PR	Mixed BMC Machinable Parcel over 0.2063 pounds
PG	5-Digit Irregular Parcel over 0.2063 pounds
PK	3-Digit Irregular Parcel over 0.2063 pounds
P5	ADC Irregular Parcel over 0.2063 pounds
P6	Mixed ADC Irregular Parcel over 0.2063 pounds

Value	Description
S	Single Piece
Z1	BMC Presort
Z2	OBMC Presort