

Mail.XML Version 26.3

System Messages Specification

Wednesday, March 20, 2024

Working Group Chair

Shawn Baldwin, WindowBook

Technical Director

Shariq Mirza, DTAC Associate, Assurety Consulting & Solutions

Editor

Shariq Mirza, DTAC Associate, Assurety Consulting & Solutions

Copyright (c) 2022 – Delivery Technology Advocacy Council (“DTAC ”). All Rights Reserved.

Mail.dat is a registered trademark of DTAC

Mail.XML is a trademark of DTAC



Copyright and Legal Notices

© 2022 Delivery Technology Advocacy Council. All Rights Reserved.

Copyright 2022 – Delivery Technology Advocacy Council (“DTAC”) is the “Copyright Owner” of “Mail.XML®”. All rights reserved by the Copyright Owner under the laws of the United States, Belgium, the European Economic Community, and all states, domestic and foreign. This document may be downloaded and copied provided that all copies retain and display the copyright and any other proprietary notices contained in this document. This document may not be sold, modified, edited, or taken out of context such that it creates a false or misleading statement or impression as to the purpose or use of the Mail.XML® specification, which is an open standard. Use of this Standard, in accord with the foregoing limited permission, shall not create for the user any rights in or to the copyright, which rights are exclusively reserved to the Copyright Owner.

DTAC and the members of the Mail.XML® Specifications - Committee (collectively and individually, "Presenters") make no representations or warranties, express or implied, including, but not limited to, warranties of merchantability, fitness for a particular purpose, title, or non-infringement. The presenters do not make any representation or warranty that the contents of this document are free from error, suitable for any purpose of any user, or that implementation of such contents will not infringe any third-party patents, copyrights, trademarks or other rights. By making use of this document, the user assumes all risks and waives all claims against Presenters.

In no event shall Presenters be liable to user (or other person) for direct, indirect, special or consequential damages arising from or related to any use of this document, including, without limitation, lost profits, business interruption, loss of programs, or other data on your information handling system even if Presenters are expressly advised of the possibility of such damages.

Some states do not allow the disclaimer or limitation of damages, so the disclaimers set forth above apply to the maximum extent permitted under applicable law.

Abstract

This document describes the messaging protocol for use by mailers and their consignees. The Mail.XML™ Transaction Protocol defines the roles and responsibilities of Shippers and Consignees and defines the format and method for message exchange. This messaging protocol is designed to be XML and Web-Services compliant.

Mail.XML and Mail.dat are trademarks of DTAC.

About Mail.XML™

Mail.XML™ is bringing a paradigm change to the industry by increasing business function specific B2B (Business to Business) communication within the industry that supports automation and in the end enables cost avoidance and higher profits through improved competence and effectiveness of communication. Mail.XML is designed to increase efficiency and lower costs by removing many manual data entry processes and enabling quick near real time communication between business partners. Mail.XML currently supports container-based scheduling, pick up and drop off business processes, as well as identifying different business entities responsible for performing different services such as quality of mailing, address correction, and delivery confirmation on a mailing. The core focus of Mail.XML is communication between industry members and from industry to the final mail processing and delivery organization that delivers the mail to the end consumer, e.g., USPS. In the next few versions of Mail.XML the focus moves across mailing supply chain channels, and includes advanced functions such as payment; automated verification; enabling first, second, and third-party communication and incorporating presort planning, printing, and distribution processes.

What's New in Mail.XML Version 26.3?

With this release, the Mail.XML Messaging Protocol moves to Version 26.3. This release supports structure changes required by mailing industry and Postal Service.

Changes supported by Mail.XML 26.3 include:

- 2619 – Proposal to support the new promotions and add-ons for 2025 Mailing Promotions.
- 2622 - Proposal to support the types of Election Mail (Election Mail Official Ballot and Election Mail Non-Ballot Materials)
- 2623 - Proposal to support Protected Origin Mixed ADC pallet preparation level for Periodicals flats.

About Mail.XML Schema Modularization

Today Mail.XML messages are grouped into 8 message types.

- Transportation Messages (TM)
- Mailing Messages (MM)
- Data Distribution Messages (DD)
- Dynamic Payment Template Messages
- Identification Messages (ID)
- Supply Chain Messages (SC)
- Informed Visibility (IV)
- System Messages
- Base: Shared simple types
- Definitions: Shared complex types and elements

The simple types shared across 2 or more modules are found in the Base schema. Likewise, the shared definitions module contains complex type definitions and elements that are shared across 2 or more modules.

Mail.XML Module Versioning Rules

The following versioning rules will be followed:

The Mail.XML wrapper schema**.xsd) will always be given the next higher version number (or Errata designation) when any update is made to base, defs or any module. The name of the .xsd file will indicate the new version and the new version number will be used in the namespace and target declarations:
xmlns:mailxml="http://delivery-tech.org/Specs/mailxml26.3/mailxml"
targetNamespace="http://delivery-tech.org/Specs/mailxml26.3/mailxml"

- When updates are made, only those modules that are updated will be given the next higher version number (or Errata letter designation).
- If updates are made to the base or defs, then the base and defs xsds will be given the next higher version number (or Errata designation) and all modules that call to them will also be given the next higher version number (or Errata designation).

For example:

- If the wrapper version is labeled as xmlns:mailxml="http://deliverytech.org/Specs/mailxml26.3A/mailxml" then at least one of the XSDs is at same version such as filename ='Mail.XML_26.3A.xsd' <- Errata A
- If the wrapper version is labeled as xmlns:mailxml="http://deliverytech.org/Specs/mailxml26.3B/mailxml" then at least one of the XSDs is at same version such as filename ='Mail.XML_26.3B.xsd' <- Errata B
- If the wrapper version is labeled as xmlns:mailxml="http://deliverytech.

org/Specs/mailxml26.3/mailxml" then at least one of the XSDs is at same version such as
filename ='Mail.XML_26.3.xsd' <- Major Version

Mail.XML 26.3 XSD Modules

The following Mail.XML XSD modules/namespaces are used:

- Mail.XML_tm.xsd: This module contains all the transportation (or FAST) messages and the attributes, elements and complex types that are unique to these messages. Namespace=Mail.XML_tm:
- Mail.XML_mm.xsd: This module contains all the mailing messages (eDoc) and the attributes, elements and complex types that are unique to these messages. Namespace=Mail.XML_mm:
- Mail.XML_iv.xsd: This module contains informed visibility messages and the attributes, elements and complex types that are unique to these messages. Namespace=Mail.XML_iv:
- Mail.XML_dd.xsd: This module contains all the data distribution messages and the attributes, elements and complex types that are unique to these messages. Namespace=Mail.XML_dd:
- Mail.XML_id.xsd: This module contains all the identification messages and the attributes, elements and complex types that are unique to these messages. Namespace=Mail.XML_id:
- Mail.XML_sc.xsd: This module contains all the supply chain messages and the attributes, elements and complex types that are unique to these messages. Namespace=Mail.XML_sc:
- Mail.XML_defs.xsd: This module contains all the common definitions of attributes, elements and complex types that are used across two or more message types. Namespace=Mail.XML_defs:
- Mail.XML_base.xsd: This module contains simple types that are shared across two or more modules that make up Mail.XML. These can be considered a building block for any message group.
Namespace=Mail.XML_base:
- Mail.XML.xsd: This module contains the system messages of Mail.XML and is used to build custom profiles for Mail.XML. Namespace=Mail.XML:

The Mail.XML™ 26.3 Messaging Documentation Set

The Mail.XML Messaging Specification has been organized into a set of documents. This *Schemas Specification* is one document in a set of documents that make up the Mail.XML Specification 26.3. Updates in this Specification are NOT backwardly compatible with previous versions. Other documents in the specification set include:

- Mail.XML™ 26.3: Transportation Messaging Specification documents all transportation messages
- Mail.XML™ 26.3: Mailing Messaging Specification documents all mailing messages
- Mail.XML™ 26.3: Informed Visibility Specification documents all informed visibility messages
- Mail.XML™ 26.3: Data Distribution Messaging Specification documents all data distribution messages
- Mail.XML™ 26.3: Identification Messaging Specification documents all identification messages
- Mail.XML™ 26.3: Supply Chain Messaging Specification documents all supply chain messages
- Mail.XML™ 26.3: System Messaging Specification documents all systems and fault messages
- Mail.XML™ 26.3: Simple Types Specification documents all simple types used across Mail.XML
- messages
- Mail.XML™ 26.3: Common Definitions Specification documents all shared elements and complex

- types.
- Mail.XML™ 26.3: Schemas contains the .XSDs that make up the Mail.XML Messaging Specification

Table of Contents

Abstract	3
About Mail.XML™	3
What's New in Mail.XML Version 26.3?	3
About Mail.XML Schema Modularization	4
Mail.XML Module Versioning Rules	4
Mail.XML 26.3 XSD Modules	5
The Mail.XML™ 26.3 Messaging Documentation Set	5
Schema mailxml_iv_26.3.xsd	8

Schema mailxml_iv_26.3.xsd

schema location: ..\XSDs\mailxml_iv_26.3.xsd
attributeFormDefault: **qualified**
elementFormDefault: **qualified**
targetNamespace: http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

Elements

[ContainerVisibilityDelivery](#)
[ContainerVisibilityEntry](#)
[ContainerVisibilityNotification](#)
[ContainerVisibilityQueryRequest](#)
[ContainerVisibilityQueryResponse](#)
[IMbMailpieceScanData](#)
[MPSVisDelivery](#)
[MPSVisNotification](#)
[MPSVisQueryRequest](#)
[MPSVisQueryResponse](#)
[StartTheClockBMEUBlock](#)
[StartTheClockDelivery](#)
[StartTheClockDropShipOrOrigin](#)
[StartTheClockNotification](#)
[StartTheClockPlantLoadBlock](#)
[StartTheClockQueryRequest](#)
[StartTheClockQueryResponse](#)

Complex types

[clockStartedType](#)
[manifestScanEventDetailType](#)
[manifestScanNotificationDataType](#)
[manifestScanQueryType](#)
[MPSNotificationDataType](#)
[MPSResponseType](#)
[MPSVisScanQueryType](#)
[PSRBlockType](#)
[scanEventQueryTypeType](#)
[ScanEventResultOptionsType](#)

Simple types

[eDocTypeType](#)
[handlingEventTypeType](#)
[mailObjectTypeType](#)
[recipientRoleType](#)
[scanEventCodeType](#)

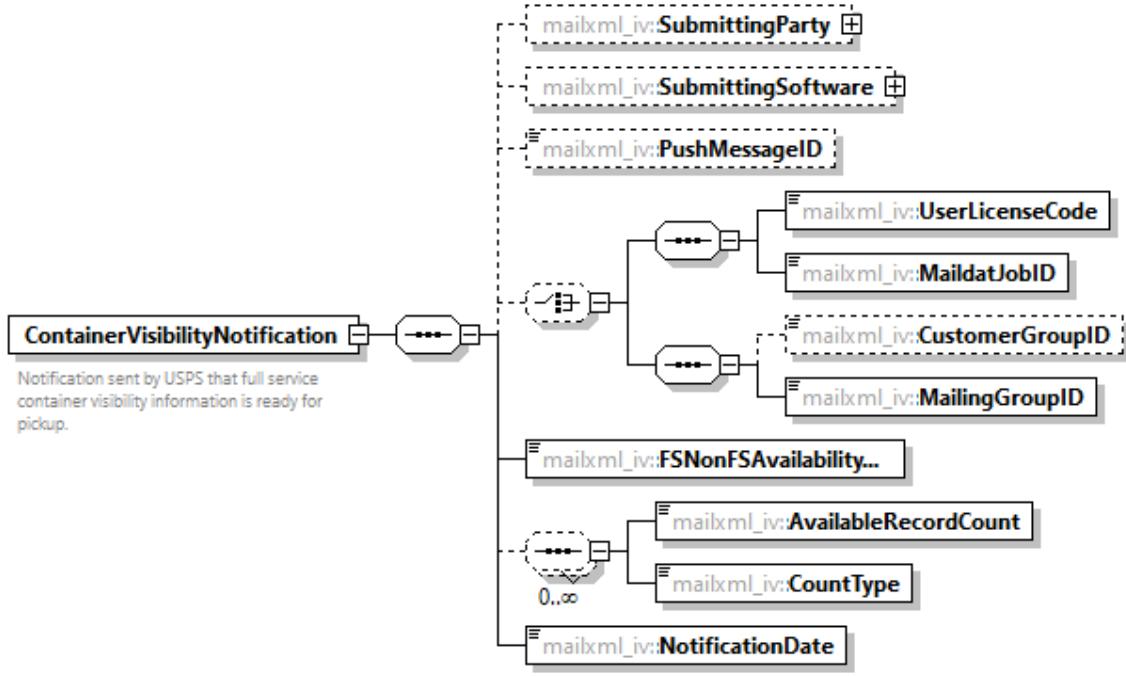
element **ContainerVisibilityDelivery**

diagram	<p>The diagram shows the ContainerVisibilityDelivery element with its attributes and associations. The element is defined as 'Delivery of full service container visibility information by uSPS.' It has an association with the grp mailxml_defs:LargeTransa... group, which contains attributes: mailxml_defs:MessageGroupID, mailxml_defs:TotalMessageCount, mailxml_defs:MessageSerialNu..., mailxml_defs:TransmittedRecor..., mailxml_defs:TotalRecordsAcro..., and mailxml_defs:LastMessage. There is also an association with the mailxml_iv:SubmittingParty and mailxml_iv:SubmittingSoftware objects, and a many-to-one association with the mailxml_defs:DataRecipient object. A note indicates that there can be multiple mailxml_iv:PushMessageID and mailxml_iv:ContainerVisibilityE... objects.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
annotation	documentation Delivery of full service container visibility information by uSPS.

element **ContainerVisibilityEntry**

diagram	<p>The diagram illustrates the structure of the ContainerVisibilityEntry element. It is defined as a class that inherits from MailObject. The class has several properties:</p> <ul style="list-style-type: none">UserLicenseCode (mailxml_iv:UserLicenseCode)MaildatJobID (mailxml_iv:MaildatJobID)CustomerGroupID (mailxml_iv:CustomerGroupID)MailingGroupID (mailxml_iv:MailingGroupID)ConsigneeApptID (mailxml_iv:ConsigneeApptID)CSAID (mailxml_iv:CSAID)ContainerID (mailxml_iv:ContainerID)MailObjectType (mailxml_iv:MailObjectType)HandlingEventType (mailxml_iv:HandlingEventType)ParentContainerID (mailxml_iv:ParentContainerID)SiblingContainerID (mailxml_iv:SiblingContainerID)ContainerType (mailxml_iv:ContainerType)ScanEvent (mailxml_iv:ScanEvent)<ul style="list-style-type: none">IMcb (mailxml_iv:IMcb)IMtb (mailxml_iv:IMtb)IMPb (mailxml_iv:IMPb)FullServiceCompliant (mailxml_iv:FullServiceCompliant)FullServiceNonCompliant (mailxml_iv:FullServiceNonCompliant)FSNonFSAvailability (mailxml_iv:FSNonFSAvailability)FullServiceIndicator (mailxml_iv:FullServiceIndicator)ScanEventCode (mailxml_iv:ScanEventCode)RecipientRole (mailxml_iv:RecipientRole)
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

element **ContainerVisibilityNotification**

diagram	 <pre> classDiagram class ContainerVisibilityNotification { <<Notification sent by USPS that full service container visibility information is ready for pickup.>> } class SubmittingParty class SubmittingSoftware class PushMessageID class UserLicenseCode class MaildatJobID class CustomerGroupID class MailingGroupID class FSNonFSAvailability... class AvailableRecordCount class CountType class NotificationDate ContainerVisibilityNotification < -- SubmittingParty ContainerVisibilityNotification < -- SubmittingSoftware ContainerVisibilityNotification < -- PushMessageID ContainerVisibilityNotification < -- UserLicenseCode ContainerVisibilityNotification < -- MaildatJobID ContainerVisibilityNotification < -- CustomerGroupID ContainerVisibilityNotification < -- MailingGroupID ContainerVisibilityNotification < -- FSNonFSAvailability... ContainerVisibilityNotification < -- AvailableRecordCount ContainerVisibilityNotification < -- CountType ContainerVisibilityNotification < -- NotificationDate </pre> <p>The diagram illustrates the structure of the ContainerVisibilityNotification element. It is a complex object composed of several sub-elements, each represented by a rectangular box with a dashed border. The sub-elements are:</p> <ul style="list-style-type: none"> SubmittingParty: Represented by <code>mailxml_iv:SubmittingParty</code>. SubmittingSoftware: Represented by <code>mailxml_iv:SubmittingSoftware</code>. PushMessageID: Represented by <code>mailxml_iv:PushMessageID</code>. UserLicenseCode: Represented by <code>mailxml_iv:UserLicenseCode</code>. MaildatJobID: Represented by <code>mailxml_iv:MaildatJobID</code>. CustomerGroupID: Represented by <code>mailxml_iv:CustomerGroupID</code>. MailingGroupID: Represented by <code>mailxml_iv:MailingGroupID</code>. FSNonFSAvailability...: Represented by <code>mailxml_iv:FSNonFSAvailability...</code>. AvailableRecordCount: Represented by <code>mailxml_iv:AvailableRecordCount</code>. CountType: Represented by <code>mailxml_iv:CountType</code>. NotificationDate: Represented by <code>mailxml_iv:NotificationDate</code>. <p>Associations between the main ContainerVisibilityNotification and its sub-elements are shown as lines connecting the objects. There is also a self-loop association on the ContainerVisibilityNotification object itself.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
annotation	<p>documentation</p> <p>Notification sent by USPS that full service container visibility information is ready for pickup.</p>

element **ContainerVisibilityQueryRequest**

diagram	<pre> classDiagram class ContainerVisibilityQueryRequest { mailxml_iv:SubmittingParty mailxml_iv:SubmittingSoftware mailxml_iv:SubmitterTrackingID mailxml_iv:UserLicenseCode mailxml_iv:MaildatJobID mailxml_iv:CustomerGroupID mailxml_iv:MailingGroupID mailxml_iv:DateRange mailxml_iv:ConsigneeApptID mailxml_iv:CSAID mailxml_iv:IMcb *0..∞ mailxml_iv:IMtb *0..∞ mailxml_iv:IMpb *0..∞ mailxml_iv:USPSFacilityLocaleKey mailxml_iv:ContainerScanState mailxml_iv:retrieveDataBy mailxml_iv:MailObjectType mailxml_iv:HandlingEventType mailxml_iv:ScanEventCode *0..∞ mailxml_iv:RecipientRole } </pre> <p>ContainerVisibilityQueryRequest</p> <p>Query request for full service container visibility information.</p> <p>Associations:</p> <ul style="list-style-type: none"> One-to-many relationship to mailxml_iv:IMcb with multiplicity 0..∞. One-to-many relationship to mailxml_iv:IMtb with multiplicity 0..∞. One-to-many relationship to mailxml_iv:IMpb with multiplicity 0..∞. <p>Attributes:</p> <ul style="list-style-type: none"> mailxml_iv:SubmittingParty mailxml_iv:SubmittingSoftware mailxml_iv:SubmitterTrackingID mailxml_iv:UserLicenseCode mailxml_iv:MaildatJobID mailxml_iv:CustomerGroupID mailxml_iv:MailingGroupID mailxml_iv:DateRange mailxml_iv:ConsigneeApptID mailxml_iv:CSAID mailxml_iv:IMcb (multiplicity 0..∞) mailxml_iv:IMtb (multiplicity 0..∞) mailxml_iv:IMpb (multiplicity 0..∞) mailxml_iv:USPSFacilityLocaleKey mailxml_iv:ContainerScanState mailxml_iv:retrieveDataBy mailxml_iv:MailObjectType mailxml_iv:HandlingEventType mailxml_iv:ScanEventCode (multiplicity 0..∞) mailxml_iv:RecipientRole
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
annotation	<p>documentation</p> <p>Query request for full service container visibility information.</p>

element **ContainerVisibilityQueryResponse**

diagram	<p>The diagram illustrates the structure of the ContainerVisibilityQueryResponse element. It starts with a main class box labeled ContainerVisibilityQueryRespo..., which has a brief description: "Response to the Query request for full service container visibility information." An association line connects this class to a large rectangular box titled attributes. Inside the attributes box, there is a group header grp mailxml_defs:LargeTransa... containing several attributes: MessageGroupID, TotalMessageCount, MessageSerialNu..., TransmittedRecor..., TotalRecordsAcro..., and LastMessage. Below this group, another association line leads to a box containing TrackingID and SubmitterTrackingID. From SubmitterTrackingID, two lines branch out: one to a box labeled QueryResults and another to a box labeled QueryError. A note at the bottom right states: "Error issued when the query data cannot be provided."</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
annotation	documentation Response to the Query request for full service container visibility information.

element **IMbMailpieceScanData**

diagram	The diagram illustrates the structure of the **IMbMailpieceScanData** element. It consists of the following components: - IMbMailpieceScanData**: The main class, represented by a rectangle with a boundary. - Attributes** (enclosed in dashed boxes): - mailxml_iv:UserLicenseCode** - mailxml_iv:MaildatJobID** - mailxml_iv:CustomerGroupID** - mailxml_iv:MailingGroupID** - Associations** (represented by lines connecting to other classes): - An association from **IMbMailpieceScanData** to **mailxml_iv:MailBundleCount** with multiplicity **1..∞**. - An association from **IMbMailpieceScanData** to **mailxml_iv:MPSCount** with multiplicity **1..∞**. - An association from **IMbMailpieceScanData** to **mailxml_iv:IMcbAndIMtbPiece...** with multiplicity **1..∞**. This class is further associated with: - mailxml_iv:IMcbPieceScanInfo** with multiplicity **1..∞** - mailxml_iv:IMtbPieceScanInfo** with multiplicity **1..∞** - mailxml_iv:IMbScanRec** with multiplicity **1..∞**

namespace http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

element **MPSVisDelivery**

diagram	<pre> classDiagram class MPSVisDelivery { <<Delivery of the Mail Piece Scanned Data to customer by USPS.>> attribute grp mailxml_defs:LargeTransa... attribute mailxml_defs:MessageGroupID attribute mailxml_defs:TotalMessageCount attribute mailxml_defs:MessageSerialNu... attribute mailxml_defs:TransmittedRecor... attribute mailxml_defs:TotalRecordsAcro... attribute mailxml_defs:LastMessage attribute mailxml_iv:SubmittingParty attribute mailxml_iv:SubmittingSoftware attribute mailxml_iv:AssumedScans attribute mailxml_iv:eDocType attribute mailxml_defs:DataRecipient attribute mailxml_iv:PushMessageID attribute mailxml_iv:IMbMailpieceScanD... attribute mailxml_iv:ManifestScanEvent attribute mailxml_iv:UnManifestedScan... } </pre> <p>The diagram shows the structure of the MPSVisDelivery element. It consists of several attributes and a sequence of events. The attributes include a group attribute 'grp mailxml_defs:LargeTransa...' containing six sub-attributes: 'MessageGroupID', 'TotalMessageCount', 'MessageSerialNu...', 'TransmittedRecor...', 'TotalRecordsAcro...', and 'LastMessage'. Below these is a sequence of events starting with 'SubmittingParty' and 'SubmittingSoftware', followed by 'AssumedScans', 'eDocType', 'DataRecipient', and 'PushMessageID'. This is followed by a sequence of events grouped under 'IMbMailpieceScanD...', which contains 'ManifestScanEvent' and 'UnManifestedScan...'. The 'IMbMailpieceScanD...' group has a multiplicity of '1..∞' at its end.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
annotation	documentation Delivery of the Mail Piece Scanned Data to customer by USPS.

element **MPSVisNotification**

diagram	<pre> graph LR MPSVisNotification[MPSVisNotification] --- SubmittingParty[mailxml_iv:SubmittingParty] MPSVisNotification --- SubmittingSoftware[mailxml_iv:SubmittingSoftware] MPSVisNotification --- AssumedScans[mailxml_iv:AssumedScans] MPSVisNotification --- eDocType[mailxml_iv:eDocType] MPSVisNotification --- DataRecipient[mailxml_defs:DataRecipient] MPSVisNotification --- PushMessageID[mailxml_iv:PushMessageID] MPSVisNotification --- UserLicenseCode[mailxml_iv:UserLicenseCode] MPSVisNotification --- MaildatJobID[mailxml_iv:MaildatJobID] MPSVisNotification --- CustomerGroupID[mailxml_iv:CustomerGroupID] MPSVisNotification --- MailingGroupID[mailxml_iv:MailingGroupID] MPSVisNotification --- AvailableRecordCount[mailxml_iv:AvailableRecordCount] MPSVisNotification --- NotificationDate[mailxml_iv:NotificationDate] MPSVisNotification --- MPSNotificationData[mailxml_iv:MPSNotificationData] MPSVisNotification --- ManifestScanNotifi... </pre> <p>Notification from USPS that the Mail Piece Scanned Data is ready.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
annotation	<p>documentation</p> <p>Notification from USPS that the Mail Piece Scanned Data is ready.</p>

element **MPSVisQueryRequest**

diagram	<pre> graph LR MPSVisQueryRequest[MPSVisQueryRequest] --- SubmittingParty[mailxml_iv:SubmittingParty] MPSVisQueryRequest --- SubmittingSoftware[mailxml_iv:SubmittingSoftware] MPSVisQueryRequest --- SubmitterTrackingID[mailxml_iv:SubmitterTrackingID] MPSVisQueryRequest --- ManifestScanQuery[mailxml_iv:ManifestScanQuery] MPSVisQueryRequest --- MPSVisScanQueryT...[mailxml_iv:MPSVisScanQueryT...] MPSVisQueryRequest --- RetrieveDataBy[mailxml_iv:RetrieveDataBy] </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

element **MPSVisQueryResponse**

diagram	<p>The diagram illustrates the structure of the MPSVisQueryResponse element. It starts with a main element box labeled "MPSVisQueryResponse". An arrow points from this box to a "attributes" section, which contains a list of attributes. The attributes are grouped under a "grp mailxml_defs:LargeTransa..." header. The listed attributes are: mailxml_defs:MessageGroupID, mailxml_defs:TotalMessageCount, mailxml_defs:MessageSerialNu..., mailxml_defs:TransmittedRecor..., mailxml_defs:TotalRecordsAcro..., and mailxml_defs:LastMessage. Below these, there is a dashed-line box containing three more attributes: mailxml_iv:TrackingID, mailxml_iv:SubmitterTrackingID, and mailxml_iv:eDocType. From the "eDocType" attribute, two arrows point to additional boxes: "mailxml_iv:QueryResults" and "mailxml_defs:QueryError". A note below states: "Error issued when the query data cannot be provided."</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

element **StartTheClockBMEUBlock**

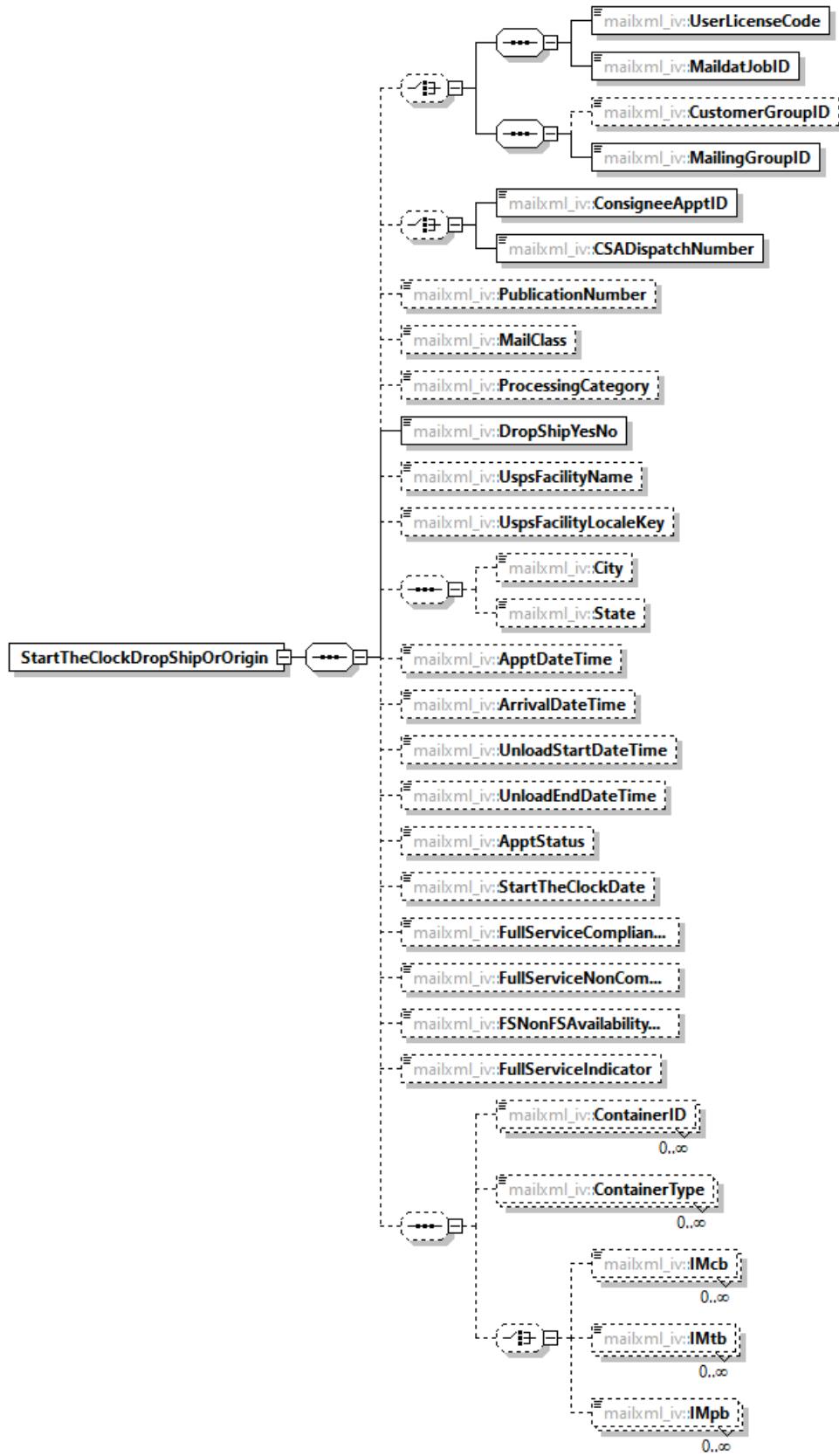
diagram	<p>The diagram shows the structure of the StartTheClockBMEUBlock element. It consists of a main element box labeled "StartTheClockBMEUBlock" followed by a sequence of three tokens (represented by ovals with three dots inside) leading to another element box labeled "mailxml_iv:StartTheClockBMEU". A multiplicity "1..∞" is shown below the second token.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

element **StartTheClockDelivery**

diagram	<p>The diagram shows a sequence of events. It begins with a participant named 'SubmittingParty' (represented by a dashed box). A synchronous message (solid arrow) labeled 'StartTheClockDelivery' is sent from the external environment to the 'SubmittingParty'. This triggers an internal process. The process involves a 'DataRecipient' (also in a dashed box) receiving a message from 'SubmittingSoftware'. The process then sends a message to 'PushMessageID'. Finally, a message is sent back to 'SubmittingParty' from 'ClockStarted', indicated by a return arrow. The multiplicity '1..∞' is shown below the return arrow.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
annotation	documentation Delivery of start the clock information to customer by USPS.

element **StartTheClockDropShipOrOrigin**

diagram



namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
-----------	---

element **StartTheClockNotification**

diagram	<pre> sequenceDiagram participant SCN as StartTheClockNotification participant SP as mailxml_iv:SubmittingParty participant SS as mailxml_iv:SubmittingSoftware participant DR as mailxml_defs:DataRecipient participant PM as mailxml_iv:PushMessageID Note over DR: Notification from USPS that start the clock information is ready to be picked up. SCN->>SP: SCN->>SS: SCN->>DR: SCN->>PM: DR->>SCN: DR->>UL: DR->>MJ: DR->>CG: DR->>MG: DR->>FNS: DR->>ND: </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
annotation	documentation Notification from USPS that start the clock information is ready to be picked up.

element **StartTheClockPlantLoadBlock**

diagram	<pre> sequenceDiagram participant SCN as StartTheClockPlantLoadBlock participant SCPLB as mailxml_iv:StartTheClockPlant... SCN->>SCPLB: SCPLB-->>SCN: 1..∞ </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

element **StartTheClockQueryRequest**

diagram	<p>The diagram illustrates the structure of the StartTheClockQueryRequest element. It starts with a central node labeled StartTheClockQueryRequest, which has two outgoing associations. The top association leads to a dashed box containing three nodes: SubmittingParty, SubmittingSoftware, and SubmitterTrackingID. The bottom association leads to another dashed box containing four nodes: ConsigneeApptID, CSADispatchNumber, UserLicenseCode, and MaildatJobID. From each of these four nodes, there is a further association leading to a third dashed box containing three nodes: CustomerGroupID, MailingGroupID, and CustomerAccount. Finally, from each of these three nodes, there is a single association leading to a fourth dashed box containing three nodes: LowerDateRange, UpperDateRange, and retrieveDataBy. Each node is represented by a rectangle with a plus sign in the top right corner.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
annotation	documentation Query request for start the clock information.

element **StartTheClockQueryResponse**

diagram	<pre> graph LR StartTheClockQueryResponse[StartTheClockQueryResponse] --- attributes[attributes] attributes --- grp["grp mailxml_defs:LargeTransa..."] attributes --- MessageGroupID["mailxml_defs:MessageGroupID"] attributes --- TotalMessageCount["mailxml_defs:TotalMessageCount"] attributes --- MessageSerialNu...["mailxml_defs:MessageSerialNu..."] attributes --- TransmittedRecor...["mailxml_defs:TransmittedRecor..."] attributes --- TotalRecordsAcro...["mailxml_defs:TotalRecordsAcro..."] attributes --- LastMessage["mailxml_defs:LastMessage"] LastMessage --- separator1[---] separator1 --- TrackingID["mailxml_iv:TrackingID"] separator1 --- SubmitterTrackingID["mailxml_iv:SubmitterTrackingID"] TrackingID --- separator2[---] separator2 --- QueryResults["mailxml_iv:QueryResults"] separator2 --- QueryError["mailxml_defs:QueryError"] </pre> <p>StartTheClockQueryResponse Response to the query for start the clock information.</p> <p>attributes</p> <ul style="list-style-type: none"> grp mailxml_defs:LargeTransa... mailxml_defs:MessageGroupID mailxml_defs:TotalMessageCount mailxml_defs:MessageSerialNu... mailxml_defs:TransmittedRecor... mailxml_defs:TotalRecordsAcro... mailxml_defs:LastMessage <p>mailxml_iv:TrackingID</p> <p>mailxml_iv:SubmitterTrackingID</p> <p>mailxml_iv:QueryResults</p> <p>mailxml_defs:QueryError</p> <p>Error issued when the query data cannot be provided.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
annotation	<p>documentation</p> <p>Response to the query for start the clock information.</p>

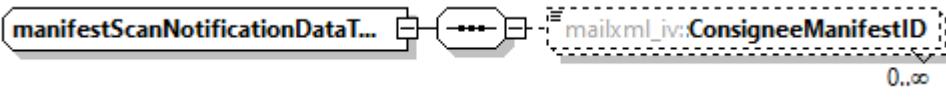
complexType **clockStartedType**

diagram	<pre> graph LR clockStartedType[clockStartedType] --- separator1[---] separator1 --- StartTheClockBME["mailxml_iv:StartTheClockBME..."] StartTheClockBME --- max1["0..∞"] StartTheClockBME --- separator2[---] separator2 --- StartTheClockDrop["mailxml_iv:StartTheClockDrop..."] StartTheClockDrop --- max2["0..∞"] StartTheClockDrop --- separator3[---] separator3 --- StartTheClockPlant["mailxml_iv:StartTheClockPlant..."] StartTheClockPlant --- max3["0..∞"] </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

complexType **manifestScanEventDetailType**

diagram	<pre> classDiagram manifestScanEventDetailType { mailxml_iv:USPSEventExtractFil... mailxml_iv:IMpb mailxml_iv:ElectronicFileNumber mailxml_iv:MailerID mailxml_iv:MailerName mailxml_iv:DestinationZipCode mailxml_iv:DestinationZipPlusF... mailxml_iv:ScanningFacilityZip mailxml_iv:ScanningFacilityName mailxml_iv:EventCode mailxml_iv:EventName mailxml_iv:EventManager mailxml_iv:EventDate mailxml_iv:EventTime mailxml_iv:MailerOwnerID mailxml_iv:CustomerReference... mailxml_iv:DestinationCountryC... mailxml_iv:RecipientName mailxml_iv:OriginalLabel mailxml_iv:UnitofMeasureCode mailxml_iv:Weight mailxml_iv:GuaranteedDelivery... mailxml_iv:GuaranteedDelivery... mailxml_iv:LogisticsManagerM... } </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

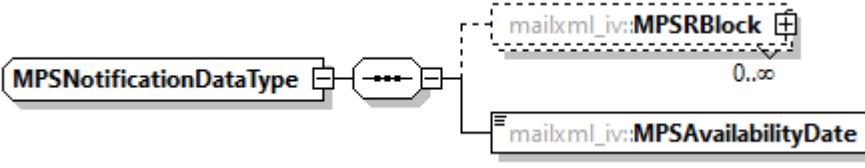
complexType **manifestScanNotificationDataType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

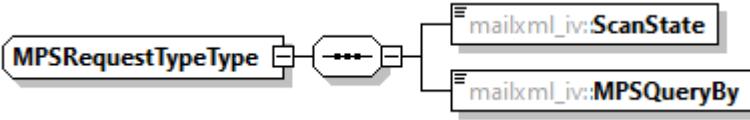
complexType **manifestScanQueryType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

complexType **MPSNotificationDataType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

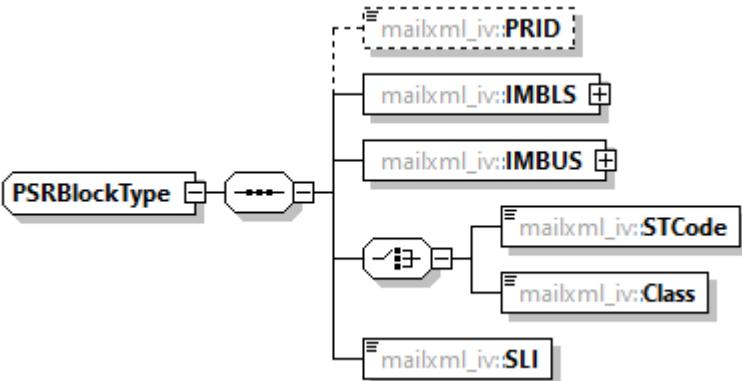
complexType **MPSRequestTypeType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

complexType **MPSVisScanQueryType**

diagram	<pre> classDiagram class MPSVisScanQueryType class UserLicenseCode class MaildatJobID class CustomerGroupID class MailingGroupID class ResultOptions class RangeLowerDate class RangeUpperDate class MID6OnThePiece class MID9OnThePiece class FacilityLocaleKey class EventType class BundleScanType class IMcb class IMtb class IMB class BundleID class FieldIncludedScanR... MPSVisScanQueryType < -- UserLicenseCode MPSVisScanQueryType < -- MaildatJobID MPSVisScanQueryType < -- CustomerGroupID MPSVisScanQueryType < -- MailingGroupID MPSVisScanQueryType *--> ResultOptions : 0..∞ MPSVisScanQueryType *--> RangeLowerDate : 0..∞ MPSVisScanQueryType *--> RangeUpperDate : 0..∞ MPSVisScanQueryType *--> MID6OnThePiece : 0..∞ MPSVisScanQueryType *--> MID9OnThePiece : 0..∞ MPSVisScanQueryType *--> FacilityLocaleKey MPSVisScanQueryType *--> EventType : 0..∞ MPSVisScanQueryType *--> BundleScanType : 0..∞ MPSVisScanQueryType *--> IMcb : 1..∞ MPSVisScanQueryType *--> IMtb : 1..∞ MPSVisScanQueryType *--> IMB : 1..∞ MPSVisScanQueryType *--> BundleID : 0..∞ MPSVisScanQueryType *--> FieldIncludedScanR... : 0..∞ </pre> <p>The diagram illustrates the structure of the MPSVisScanQueryType complex type. It consists of several components:</p> <ul style="list-style-type: none"> Root Element: MPSVisScanQueryType Attributes: <ul style="list-style-type: none"> UserLicenseCode MaildatJobID CustomerGroupID MailingGroupID Associations: <ul style="list-style-type: none"> ResultOptions (multiplicity 0..∞) RangeLowerDate (multiplicity 0..∞) RangeUpperDate (multiplicity 0..∞) MID6OnThePiece (multiplicity 0..∞) MID9OnThePiece (multiplicity 0..∞) FacilityLocaleKey EventType (multiplicity 0..∞) BundleScanType (multiplicity 0..∞) IMcb (multiplicity 1..∞) IMtb (multiplicity 1..∞) IMB (multiplicity 1..∞) BundleID (multiplicity 0..∞) FieldIncludedScanR... (multiplicity 0..∞)
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

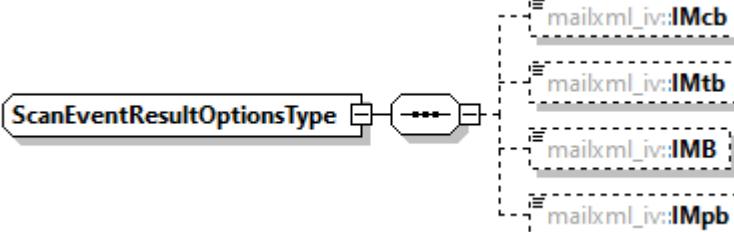
complexType **PSRBlockType**

diagram	 <p>The diagram illustrates the structure of the PSRBlockType complex type. It begins with a rounded rectangle labeled "PSRBlockType". A sequence of three association ends follows, leading to a dashed box labeled "mailxml_iv:PRID". From "PRID", two associations branch out to "mailxml_iv:IMBLS" and "mailxml_iv:IMBUS", each ending in a plus sign (+). Another association leads from "PRID" to a composite structure containing "mailxml_iv:STCode" and "mailxml_iv:Class", also ending in a plus sign (+). Finally, an association leads to "mailxml_iv:SLI".</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

complexType **scanEventQueryType**

diagram	<pre> classDiagram class scanEventQueryType { <<mailxml_iv>> UserLicenseCode MaildatJobID CustomerGroupID MailingGroupID ResultOptions 0..∞ RangeLowerDate RangeUpperDate MID6OnThePiece MID9OnThePiece FacilityLocaleKey EventType 0..∞ BundleScanType 0..∞ IMcb 1..∞ IMtb 1..∞ IMB 1..∞ BundleID 0..∞ FieldIncludedScanR... 0..∞ } class scanEventQueryType { <<mailxml_iv>> ... } scanEventQueryType < --> scanEventQueryType </pre> <p>The diagram illustrates the structure of the scanEventQueryType complex type. It features a primary sequence of elements: UserLicenseCode, MaildatJobID, CustomerGroupID, MailingGroupID, ResultOptions (with multiplicity 0..∞), RangeLowerDate, RangeUpperDate, MID6OnThePiece, MID9OnThePiece, FacilityLocaleKey, EventType (with multiplicity 0..∞), BundleScanType (with multiplicity 0..∞), IMcb (with multiplicity 1..∞), IMtb (with multiplicity 1..∞), IMB (with multiplicity 1..∞), BundleID (with multiplicity 0..∞), and FieldIncludedScanR... (with multiplicity 0..∞). A dashed line indicates a self-relation for the scanEventQueryType element.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

complexType ScanEventResultOptionsType

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv

simpleType eDocTypeType

namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
type	restriction of xs:string

simpleType handlingEventTypeType

namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
type	restriction of xs:string

simpleType mailObjectTypeType

namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
type	restriction of xs:string

simpleType recipientRoleType

namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
type	restriction of xs:string

simpleType scanEventCodeType

namespace	http://delivery-tech.org/Specs/mailxml26.3/mailxml_iv
type	restriction of xs:string