

Mail.XML Version 26.2

System Messages Specification

Monday, September 11, 2023

Working Group Chair

Shawn Baldwin, BCC Software

Technical Director

Shariq Mirza, DTAC Associate, Assurety Consulting & Solutions

Editor

Shariq Mirza, DTAC Associate, Assurety Consulting & Solutions

Copyright (c) 2023 – 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

© 2023 Delivery Technology Advocacy Council. All Rights Reserved.

Copyright 2023 – 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 26.2?

With this release, the Mail.XML Messaging Protocol moves to Version 26.2. This release supports structure changes required by mailing industry and Postal Service.

Changes supported by Mail.XML 26.2 include:

- CR 2615 - Updating characteristicIncentiveType and adding FG for First-Class Growth Incentive Credit Redemption
- CR 2616 - Updating characteristicIncentiveType and adding MG for USPS Marketing Mail Growth Incentive Credit Redemption
- CR 2617 - Updating containerLevelType, and adding AU for Protected Mixed ADC and AV for Protected Mixed NDC

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/mailxml26.2/mailxml"
targetNamespace="http://delivery-tech.org/Specs/mailxml26.2/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/mailxml26.2A/mailxml" then at least one of the XSDs is at same version such as filename ='Mail.XML_26.2A.xsd' <- Errata A
- If the wrapper version is labeled as xmlns:mailxml="http://deliverytech.org/Specs/mailxml26.2B/mailxml" then at least one of the XSDs is at same version such as filename ='Mail.XML_26.2B.xsd' <- Errata B

- If the wrapper version is labeled as xmlns:mailxml="http://deliverytech.org/Specs/mailxml26.2/mailxml" then at least one of the XSDs is at same version such as filename = 'Mail.XML_26.2.xsd' <- Major Version

Mail.XML 26.2 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™ 26.2 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 26.2. Updates in this Specification are NOT backwardly compatible with previous versions. Other documents in the specification set include:

- Mail.XML™ 26.2: Transportation Messaging Specification documents all transportation messages
- Mail.XML™ 26.2: Mailing Messaging Specification documents all mailing messages
- Mail.XML™ 26.2: Informed Visibility Specification documents all informed visibility messages
- Mail.XML™ 26.2: Data Distribution Messaging Specification documents all data distribution messages
- Mail.XML™ 26.2: Identification Messaging Specification documents all identification messages
- Mail.XML™ 26.2: Supply Chain Messaging Specification documents all supply chain messages
- Mail.XML™ 26.2: System Messaging Specification documents all systems and fault messages
- Mail.XML™ 26.2: Simple Types Specification documents all simple types used across Mail.XML messages

- Mail.XML™ 26.2: Common Definitions Specification documents all shared elements and complex
- types.
- Mail.XML™ 26.2: 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 26.2?	3
About Mail.XML Schema Modularization	4
Mail.XML Module Versioning Rules	4
Mail.XML 26.2 XSD Modules	5
The Mail.XML™ 26.2 Messaging Documentation Set	5
Schema mailxml_defs_26.2.xsd	8

Schema mailxml_defs_26.2.xsd

schema location: C:\Users\NabilRahman\Desktop\XML SCHEMA UPDATE\26.2\Mail.XML 26.2\MailXML 26.2\XSDs\mailxml_defs_26.2.xsd
attributeFormDefault: **qualified**
elementFormDefault: **qualified**
targetNamespace: **http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs**

Elements

[ContainerDetailData](#)
[ContainerInfoData](#)
[DataRecipient](#)
[ExtraServiceData](#)
[LargeTransactionDivider](#)
[LargeTransactionDividerResult](#)
[QueryError](#)

Complex types

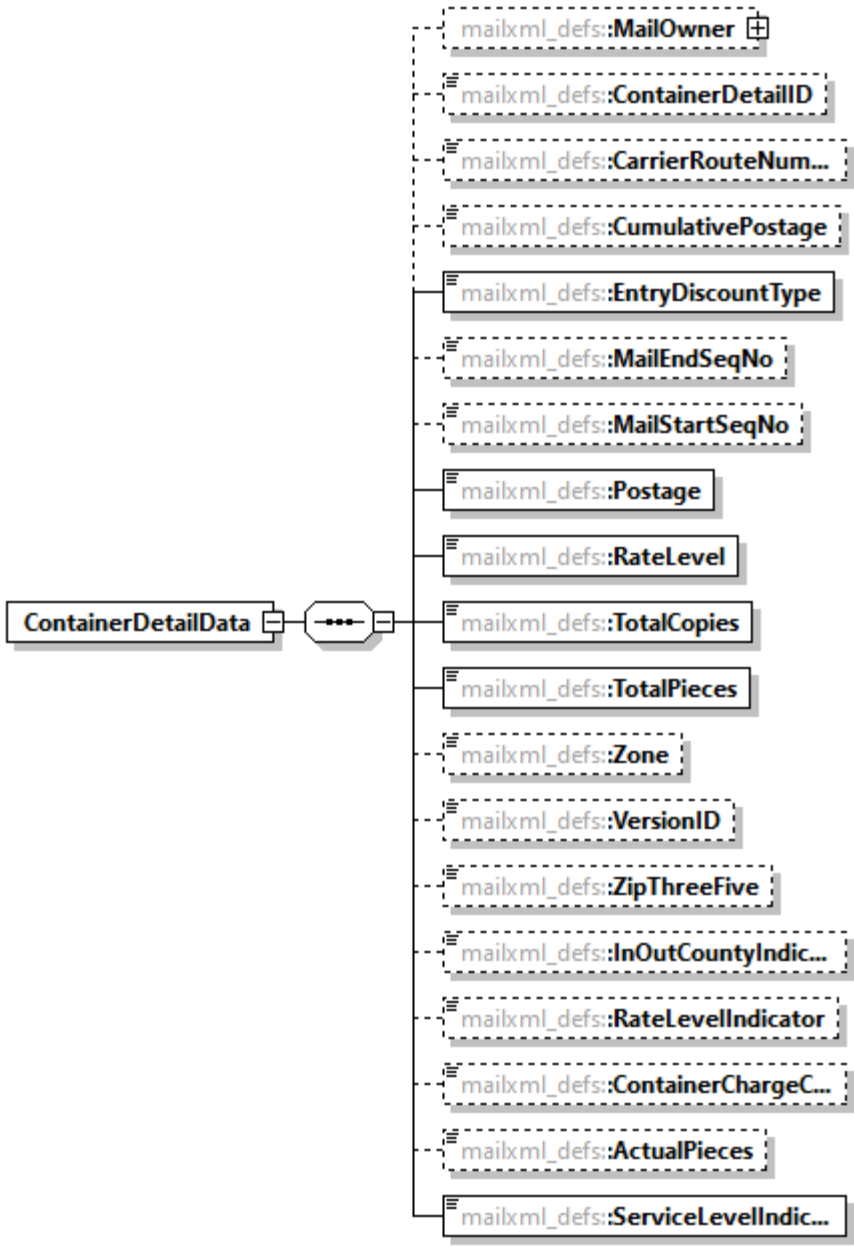
[addressType](#)
[basicReturnInfoType](#)
[commonContactType](#)
[consigneeFacilityType](#)
[consolidatorCommunicationInfoType](#)
[contactIDType](#)
[ContainerDiscrepancyType](#)
[containerErrorWarningBlockType](#)
[containerIDType](#)
[ContainerInductedType](#)
[containerInfoDataType](#)
[containerKeysInfoType](#)
[containerPostInductionInfoType](#)
[containerPreInductionInfoType](#)
[containerReleaseInfoType](#)
[containerStatusInfoType](#)
[contentIDType](#)
[CSQContainerDetailDataType](#)
[CSQContainerInfoDataType](#)
[CSQlinkingContainerIDType](#)
[documentVersionDataforCSQType](#)
[errorWarningType](#)
[fullContainerIDType](#)
[gPSCoordinates](#)
[IMcbAndIMtbPieceScanInfoType](#)
[IMcbPieceScanInfoType](#)
[IMtbPieceScanInfoType](#)
[inductionCloseoutInfoType](#)
[inductionIssuesType](#)
[InductionProblemType](#)
[intelligentMailBarcodeType](#)
[intelligentMailPackageBarcodeType](#)
[maildatContainerIDType](#)

Simple types

[bundleScanTypeType](#)
[containerDiscrepancyCategoryType](#)
[containerScanStateType](#)
[containerStatusType](#)
[countTypeType](#)
[einductionDataSourceType](#)
[fullServiceComplianceIndicatorType](#)
[MPSQueryType](#)
[MPSSStateType](#)
[pieceScanEventTypeType](#)
[problemCategoryType](#)
[problemTypeType](#)
[reasonCodeType](#)
[retrieveDataByType](#)

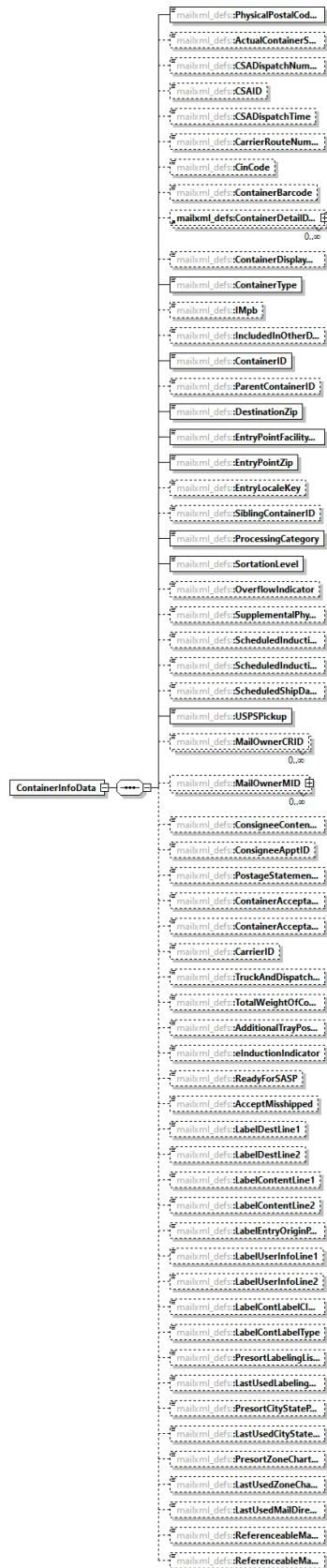
[MaildatMPUInfoType](#)
[mailPieceIDType](#)
[MailXMLContainerIDType](#)
[mailxmlDetailType](#)
[measurementType](#)
[MIDType](#)
[MPSRBlockType](#)
[MPSVisIncludedInScanRecFlagType](#)
[MPSVisResultOptionsType](#)
[palletHandoffInfoType](#)
[participantIDType](#)
[permitPublicationDataType](#)
[pickupApptBlockResponseType](#)
[postageStatementSummaryType](#)
[PSRBlockType](#)
[queryErrorType](#)
[returnInfoType](#)
[scanEventType](#)
[submittingSoftwareType](#)
[zipCode](#)

element **ContainerDetailData**

<p>diagram</p>	 <pre> classDiagram class ContainerDetailData { MailOwner ContainerDetailID CarrierRouteNum... CumulativePostage EntryDiscountType MailEndSeqNo MailStartSeqNo Postage RateLevel TotalCopies TotalPieces Zone VersionID ZipThreeFive InOutCountyIndic... RateLevelIndicator ContainerChargeC... ActualPieces ServiceLevelIndic... } </pre> <p>The diagram illustrates the structure of the ContainerDetailData element. It is a container element, indicated by the square symbol with a small circle on its right side. A dashed line connects it to a circle containing three dots, which then branches out to a vertical list of 17 child elements. Each child element is represented by a rectangular box with a small square icon on its left side. The elements are as follows:</p> <ul style="list-style-type: none"> mailxml_defs:MailOwner (with a small square icon containing a plus sign) mailxml_defs:ContainerDetailID mailxml_defs:CarrierRouteNum... mailxml_defs:CumulativePostage mailxml_defs:EntryDiscountType mailxml_defs:MailEndSeqNo mailxml_defs:MailStartSeqNo mailxml_defs:Postage mailxml_defs:RateLevel mailxml_defs:TotalCopies mailxml_defs:TotalPieces mailxml_defs:Zone mailxml_defs:VersionID mailxml_defs:ZipThreeFive mailxml_defs:InOutCountyIndic... mailxml_defs:RateLevelIndicator mailxml_defs:ContainerChargeC... mailxml_defs:ActualPieces mailxml_defs:ServiceLevelIndic...
<p>namespace</p>	<p>http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs</p>

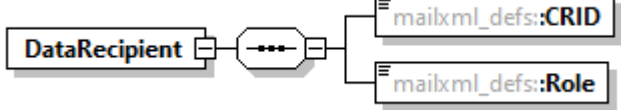
element **ContainerInfoData**

diagram

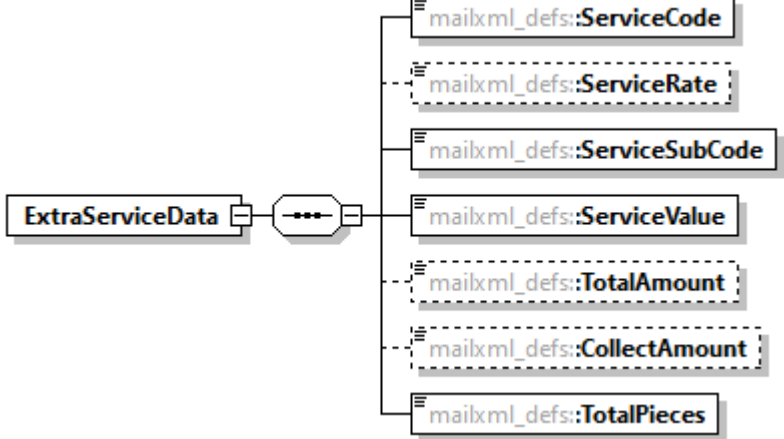


namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
-----------	---


element **DataRecipient**

diagram	 <pre> graph LR DR[DataRecipient] --- C1(()) C1 --- CRID[mailxml_defs::CRID] C1 --- Role[mailxml_defs::Role] </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

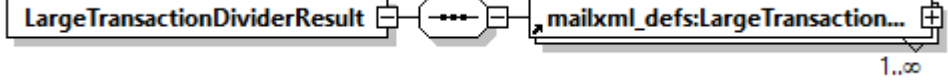
element **ExtraServiceData**

diagram	 <pre> graph LR ESD[ExtraServiceData] --- C1(()) C1 --- SC[mailxml_defs::ServiceCode] C1 --- SR[mailxml_defs::ServiceRate] C1 --- SSC[mailxml_defs::ServiceSubCode] C1 --- SV[mailxml_defs::ServiceValue] C1 --- TA[mailxml_defs::TotalAmount] C1 --- CA[mailxml_defs::CollectAmount] C1 --- TP[mailxml_defs::TotalPieces] </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

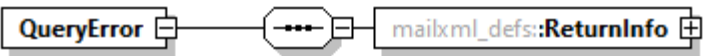
element **LargeTransactionDivider**

diagram	 <pre> graph LR LTD[LargeTransactionDivider] --- C1(()) C1 --- TI[mailxml_defs::TrackingID] </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

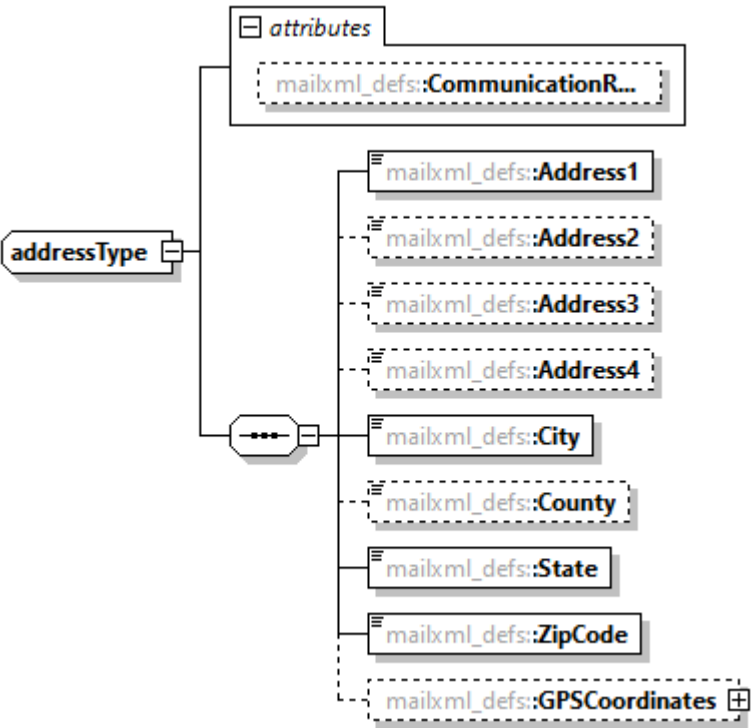
element **LargeTransactionDividerResult**

diagram	 <pre> graph LR LTRD[LargeTransactionDividerResult] --- C1(()) C1 --- LTR[mailxml_defs::LargeTransaction...] LTR -- 1..∞ --> LTR </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

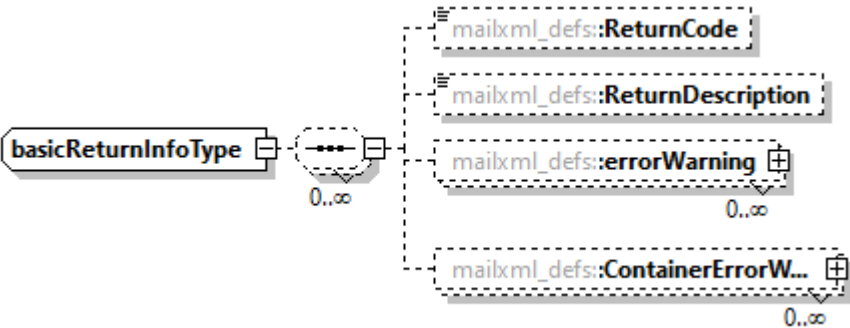
element **QueryError**

diagram	 <p>The diagram shows the QueryError element connected to the mailxml_defs::ReturnInfo element via a dashed line with a circle containing three dots. Below the diagram, the text reads: "Error issued when the query data cannot be provided."</p>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
annotation	<p>documentation</p> <p>Error issued when the query data cannot be provided.</p>

complexType **addressType**

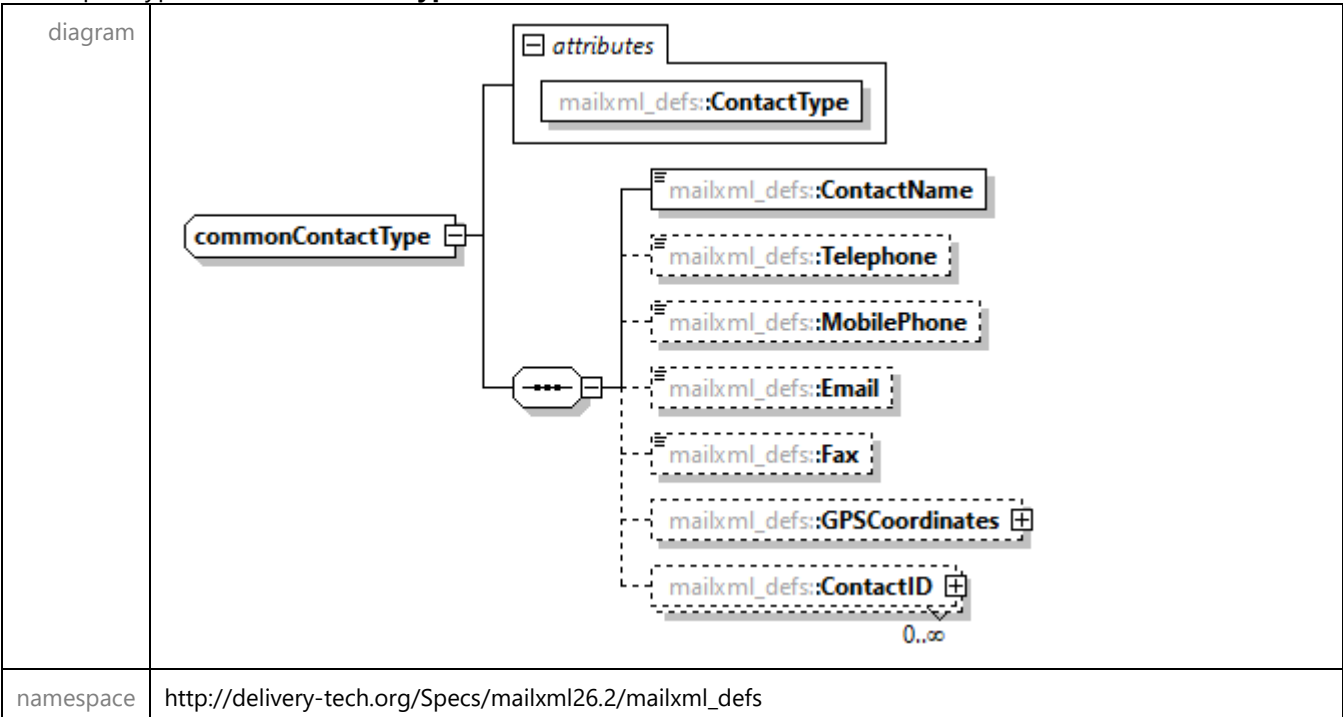
diagram	 <p>The diagram shows the addressType complex type. It has an attributes section containing mailxml_defs::CommunicationR.... The main body of the type contains a sequence of elements: mailxml_defs::Address1, mailxml_defs::Address2, mailxml_defs::Address3, mailxml_defs::Address4, mailxml_defs::City, mailxml_defs::County, mailxml_defs::State, mailxml_defs::ZipCode, and mailxml_defs::GPSCoordinates. The first four address elements are connected to a dashed line with a circle containing three dots, indicating a sequence. The GPSCoordinates element is connected to the same sequence via a dashed line with a circle containing three dots and a plus sign, indicating it is optional.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **basicReturnInfoType**

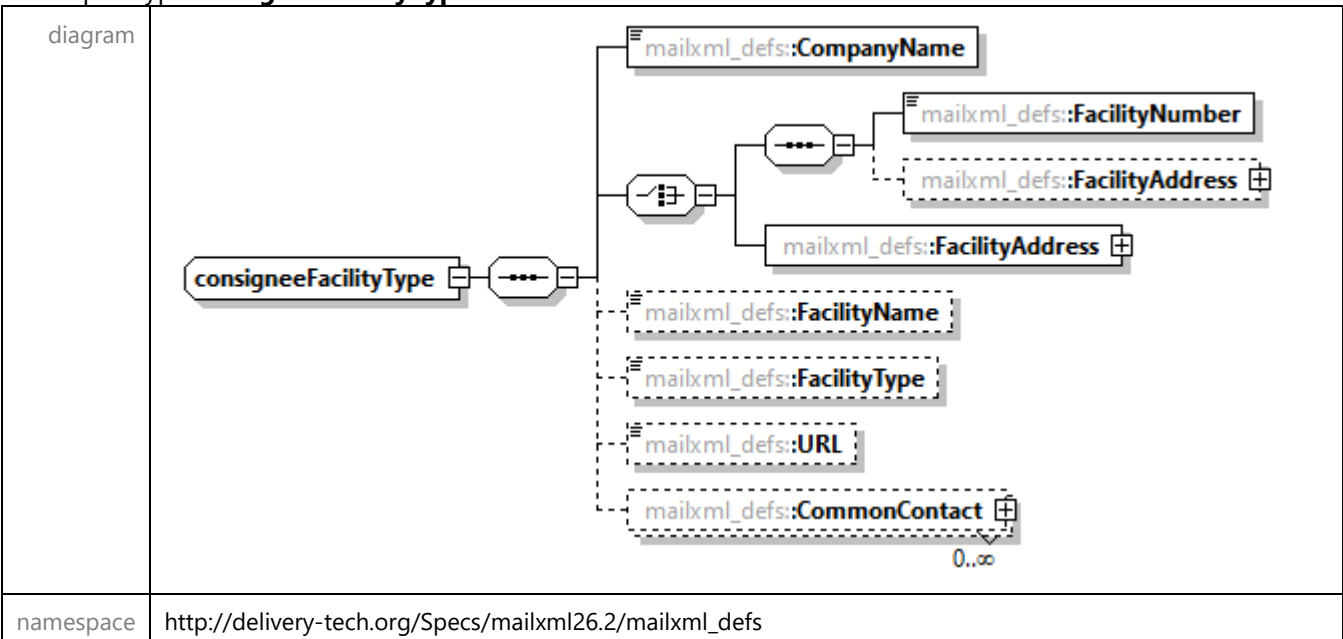
diagram	 <p>The diagram shows the basicReturnInfoType complex type. It contains a sequence of elements: mailxml_defs::ReturnCode, mailxml_defs::ReturnDescription, mailxml_defs::errorWarning, and mailxml_defs::ContainerErrorW.... The first two elements are connected to a dashed line with a circle containing three dots, indicating a sequence. The errorWarning and ContainerErrorW... elements are connected to the same sequence via dashed lines with a circle containing three dots and a plus sign, indicating they are optional. The cardinality 0..∞ is shown below the sequence line.</p>
---------	---

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
-----------	---

complexType **commonContactType**



complexType **consigneeFacilityType**



complexType **consolidatorCommunicationInfoType**

diagram

consolidatorCommunicationInf...

```

<?xml version="1.0" encoding="UTF-8" ?>
<consolidatorCommunicationInfoType>
  <PalletID>
  </PalletID>
  <ContainerBarcode>
  </ContainerBarcode>
  <InterrollID>
  </InterrollID>
  <BillingNumber>
  </BillingNumber>
  <OriginFacilityInfo>
  </OriginFacilityInfo>
  <ConsolidatorFacility>
  </ConsolidatorFacility>
  <PalletCount>
  </PalletCount>
  <FacilityType>
  </FacilityType>
  <DestinationCity>
  </DestinationCity>
  <DestinationState>
  </DestinationState>
  <DestinationZipCode>
  </DestinationZipCode>
  <LocalKey>
  </LocalKey>
  <FirstPickupDate>
  </FirstPickupDate>
  <ActualWarehouse>
  </ActualWarehouse>
  <DueAtConsolidator>
  </DueAtConsolidator>
  <RecdAtConsolidator>
  </RecdAtConsolidator>
  <ConsolidatedInvoice>
  </ConsolidatedInvoice>
  <ShippedQuantity>
  </ShippedQuantity>
  <ShipmentComplete>
  </ShipmentComplete>
  <ConsolidatorShip>
  </ConsolidatorShip>
  <BeginDeliveryWindow>
  </BeginDeliveryWindow>
  <EndDeliveryWindow>
  </EndDeliveryWindow>
  <ApptDate>
  </ApptDate>
  <ActualDeliveryDate>
  </ActualDeliveryDate>
  <CoPalPoolDate>
  </CoPalPoolDate>
  <DeliveryType>
  </DeliveryType>
  <PurchaseOrderNumber>
  </PurchaseOrderNumber>
  <SackTrayCount>
  </SackTrayCount>
  <PieceCount>
  </PieceCount>
  <Weight>
  </Weight>
  <ContainerStatus>
  </ContainerStatus>
  <LastPickupDate>
  </LastPickupDate>
  <LastPickupType>
  </LastPickupType>
  <CurrentZone>
  </CurrentZone>
  <CurrentLocation>
  </CurrentLocation>
  <AppointmentDate>
  </AppointmentDate>
  <AppointmentNumber>
  </AppointmentNumber>
  <CustomerName>
  </CustomerName>
  <InvoiceNumber>
  </InvoiceNumber>
  <BillTo>
  </BillTo>
  <MailPreparerJobID>
  </MailPreparerJobID>
  <StatComment>
  </StatComment>
  <ServiceRequestID>
  </ServiceRequestID>
  <ServiceItem>
  </ServiceItem>
  <ReferenceNumber>
  </ReferenceNumber>
  <VersionNumber>
  </VersionNumber>
  <Method>
  </Method>
  <OutboundShipping>
  </OutboundShipping>
  <MiddleDayOfTransit>
  </MiddleDayOfTransit>
  <ActualTransit>
  </ActualTransit>
  <HoldDays>
  </HoldDays>
  <DeliveryConfirmation>
  </DeliveryConfirmation>
  <ScheduledPickup>
  </ScheduledPickup>
  <ScheduledPickup>
  </ScheduledPickup>
  <PS8125Received>
  </PS8125Received>
  <SegmentTapeCode>
  </SegmentTapeCode>
  <BillingVersionNumber>
  </BillingVersionNumber>
  <MailPreparerJobNumber>
  </MailPreparerJobNumber>
  <MailPreparerJobNumber>
  </MailPreparerJobNumber>
  <StartFacilityDate>
  </StartFacilityDate>
  <EndFacilityDate>
  </EndFacilityDate>
  <DSMSStatus>
  </DSMSStatus>
  <ReleasedBy>
  </ReleasedBy>
  <ReleasedDate>
  </ReleasedDate>
</consolidatorCommunicationInfoType>

```


namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
-----------	---

complexType **contactIDType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

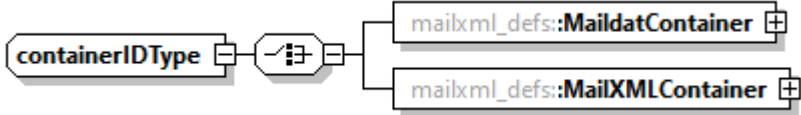
complexType **ContainerDiscrepancyType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

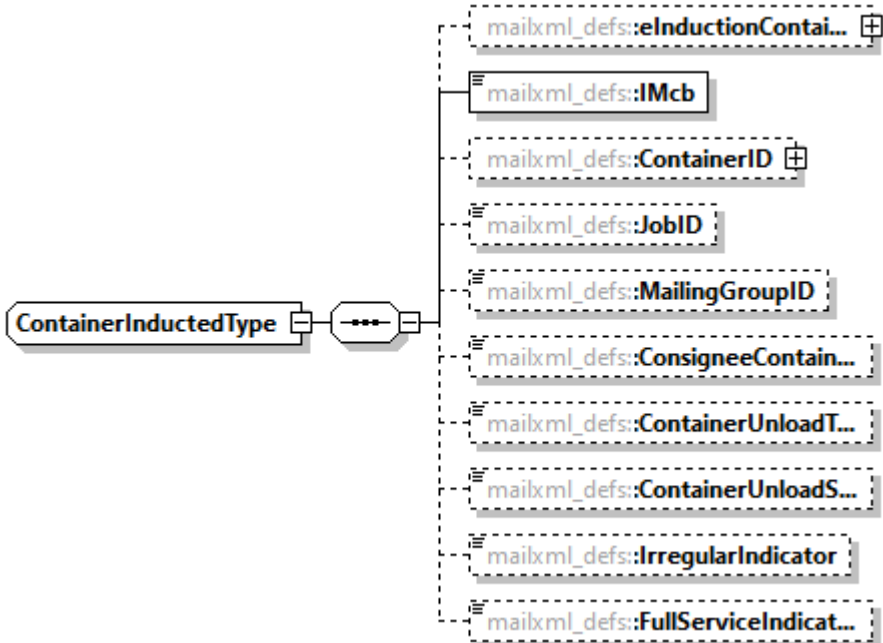
complexType **containerErrorWarningBlockType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **containerIDType**

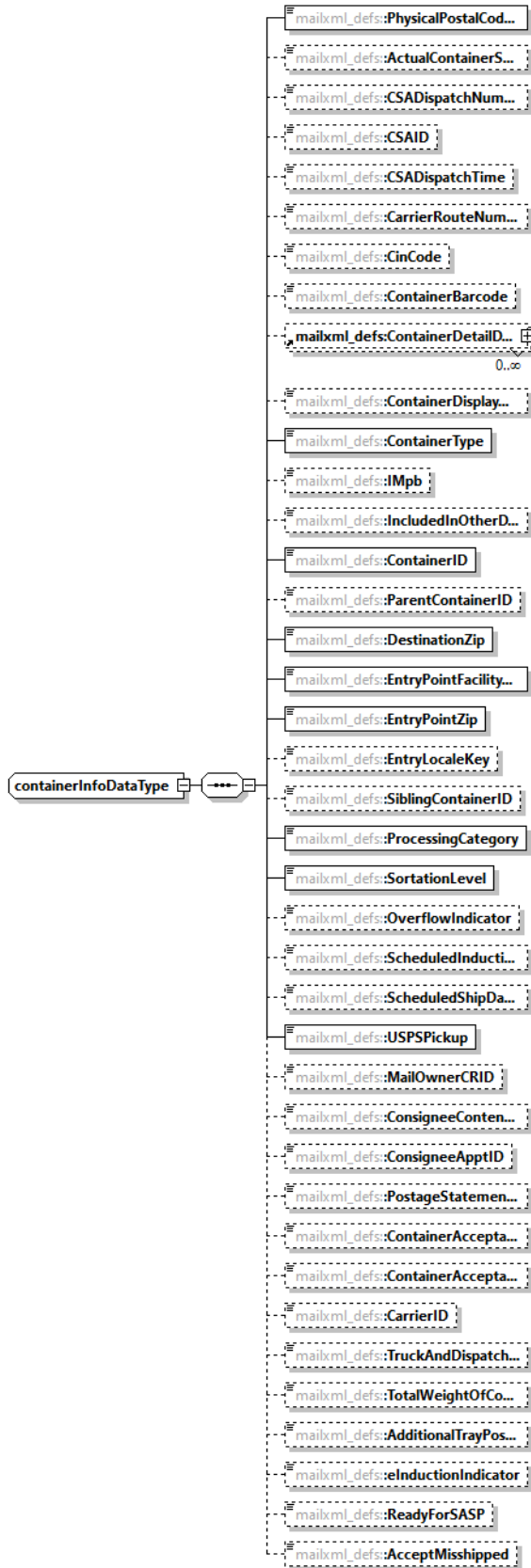
diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **ContainerInductedType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **containerInfoDataType**

diagram



namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
-----------	---

complexType **containerKeysInfoType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **containerPostInductionInfoType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **containerPreInductionInfoType**

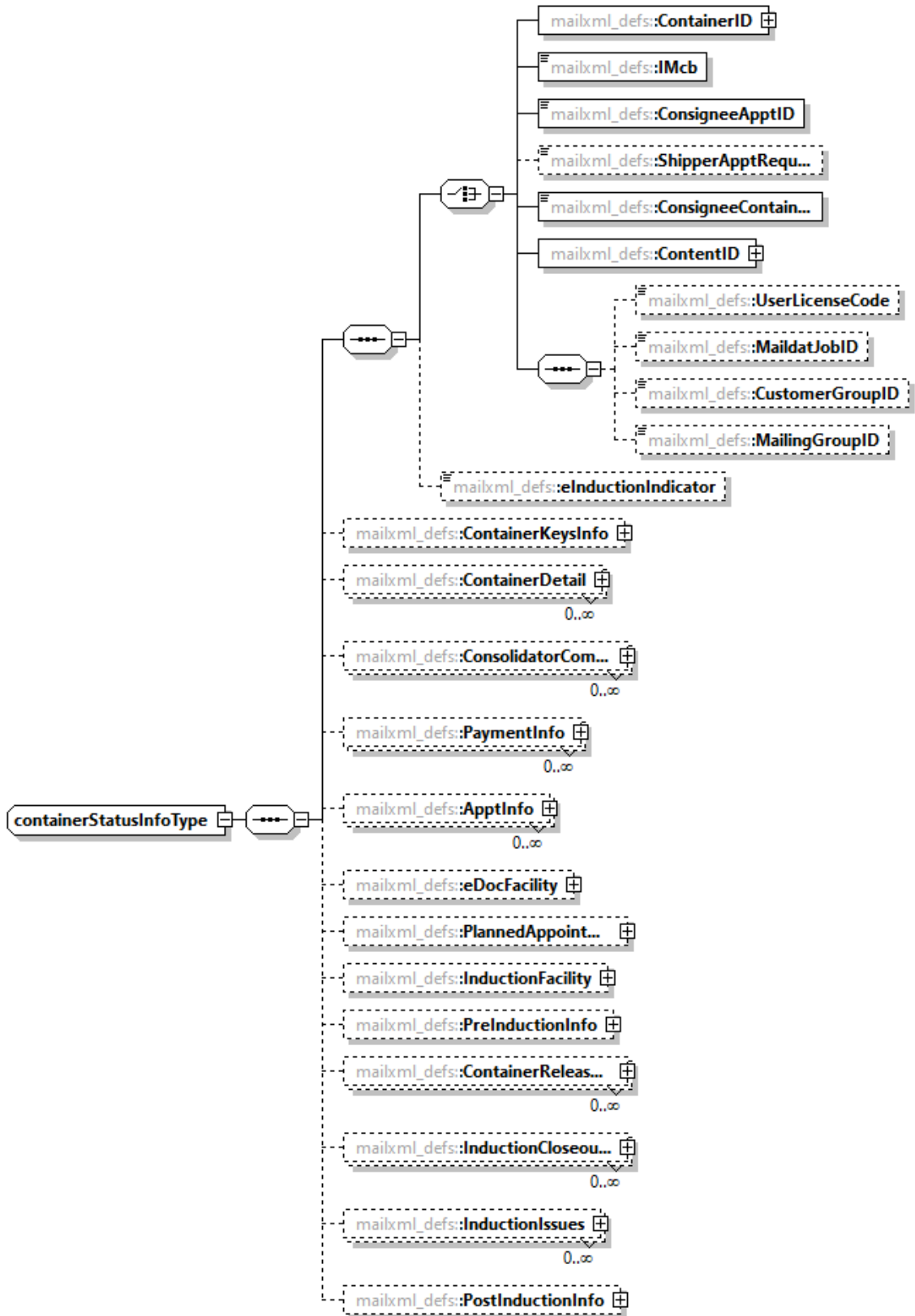
diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **containerReleaseInfoType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **containerStatusInfoType**

diagram



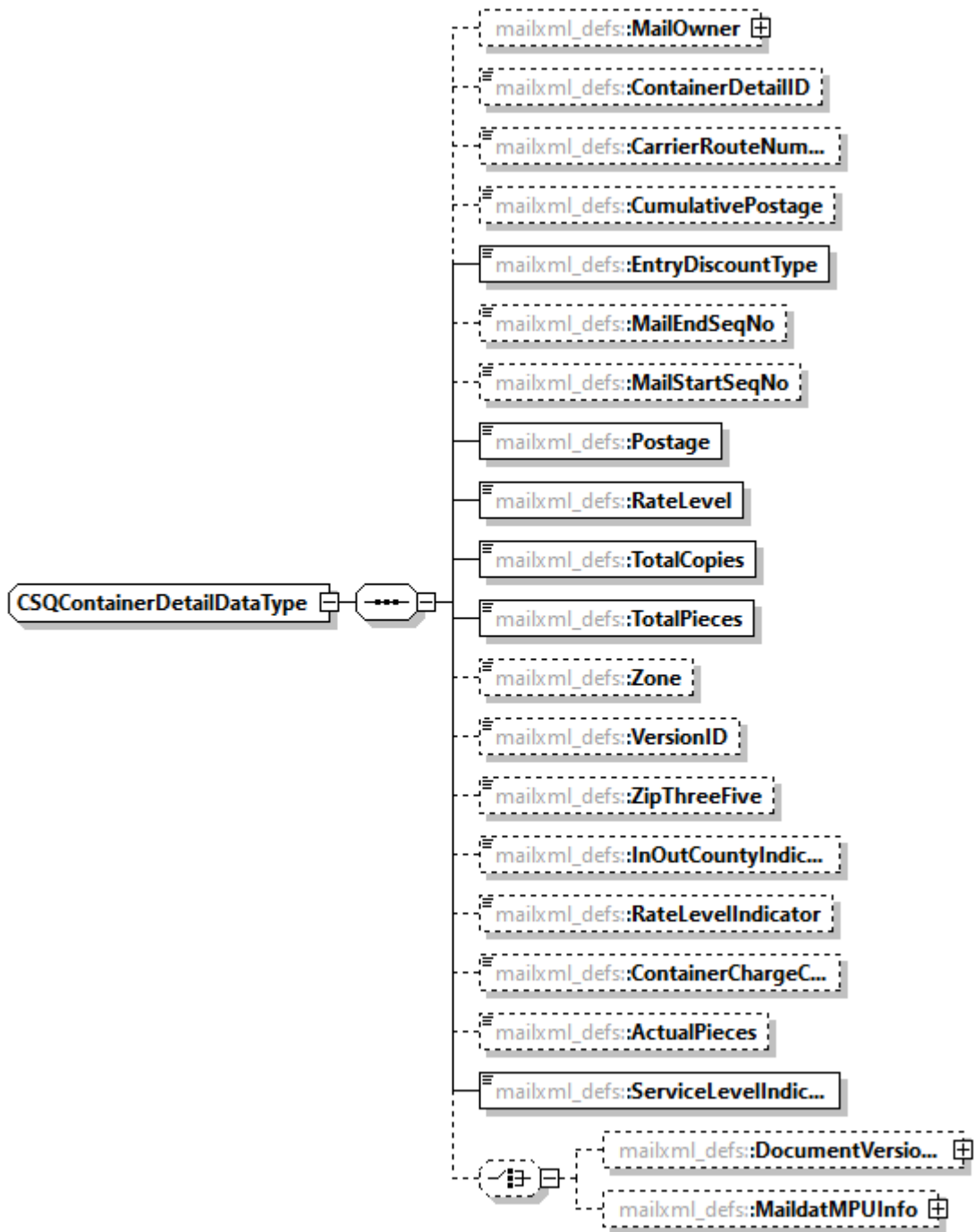
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
-----------	---

complexType **contentIDType**

diagram	<p>The diagram shows a complex type named contentIDType. It has a single attribute named <i>attributes</i>. The attribute's value is a choice between two elements: <code>mailxml_defs:ConsigneeConten...</code> (solid border) and <code>mailxml_defs:SchedulerContent...</code> (dashed border).</p>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **CSQContainerDetailDataType**

diagram



namespace http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

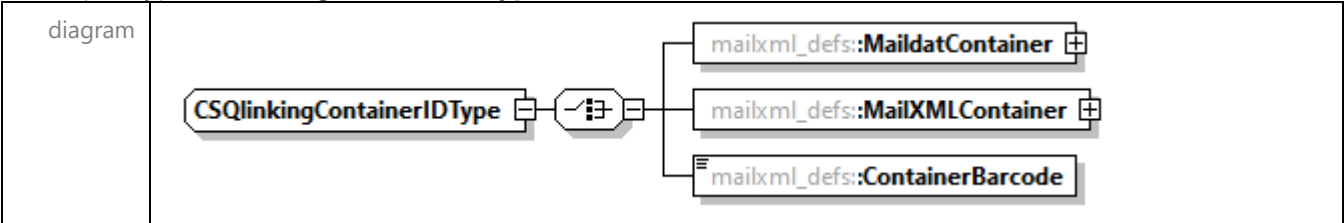
complexType **CSQcontainerInfoDataType**

diagram



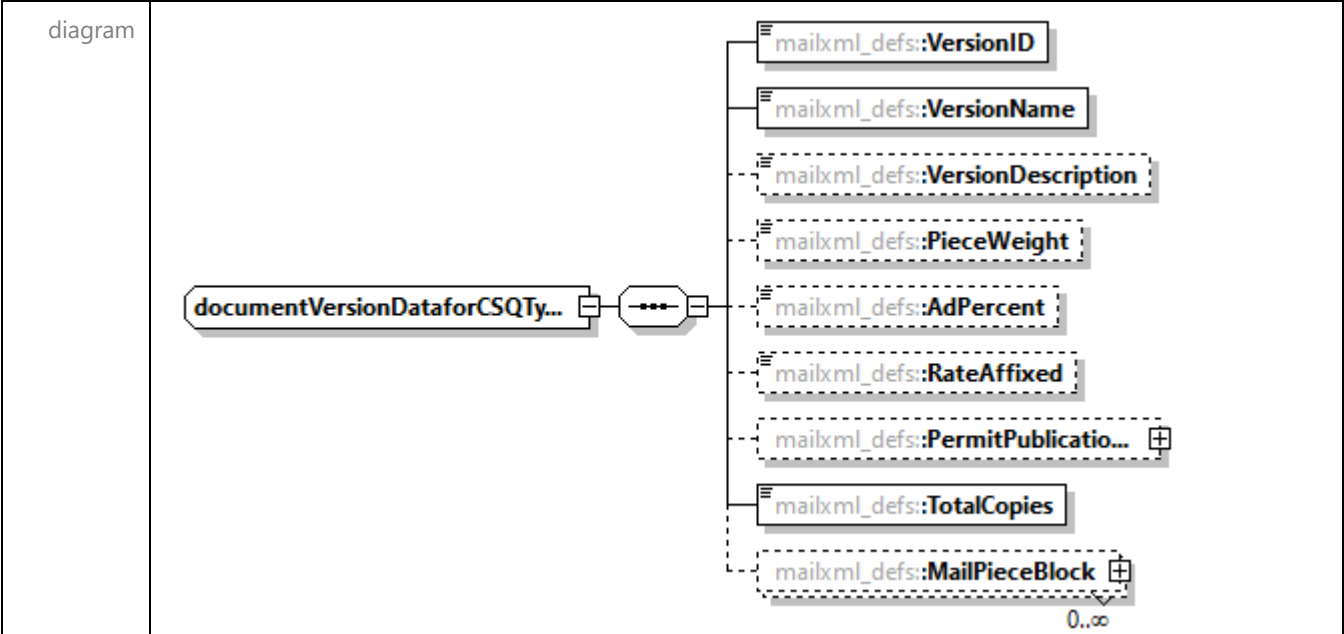
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
-----------	---

complexType **CSQlinkingContainerIDType**



namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
-----------	---

complexType **documentVersionDataforCSQType**



namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
-----------	---

complexType **errorWarningType**

diagram	<pre> xsd:complexType base="errorWarningType" sequence element Code element Severity element KeyIDValues element SubmissionDate element ErrorDescription element Action element TrackingID element SubmitterTrackin... element MailxmlDetail </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **fullContainerIDType**

diagram	<pre> xsd:complexType base="fullContainerIDType" sequence element ContainerID element IMtb element IMcb </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **gPSCoordinates**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **IMcbAndIMtbPieceScanInfoType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **IMcbPieceScanInfoType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **IMtbPieceScanInfoType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **inductionCloseoutInfoType**

diagram



namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
-----------	---

complexType **inductionIssuesType**

diagram	<p>The diagram shows the inductionIssuesType complex type. It consists of a sequence of three elements: mailxml_defs:ContainerProblem, mailxml_defs:AppointmentPro..., and mailxml_defs:TruckLoadProblem. Each element is optional (indicated by a dashed border) and can occur 0 or more times (indicated by $0..∞$).</p>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **InductionProblemType**

diagram	<p>The diagram shows the InductionProblemType complex type. It consists of a sequence of five elements: mailxml_defs:ProblemID, mailxml_defs:ProblemDescripti..., mailxml_defs:ProblemType, mailxml_defs:ProblemCategory, and mailxml_defs:ProblemResolution. The first element is required (solid border), while the others are optional (dashed border).</p>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

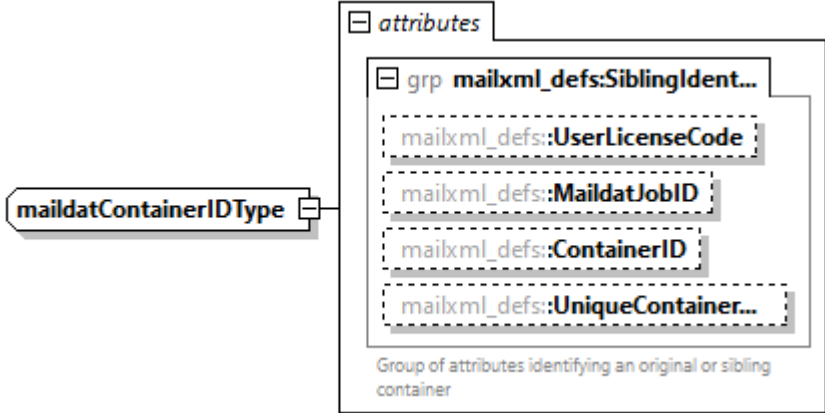
complexType **intelligentMailBarcodeType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **intelligentMailPackageBarcodeType**

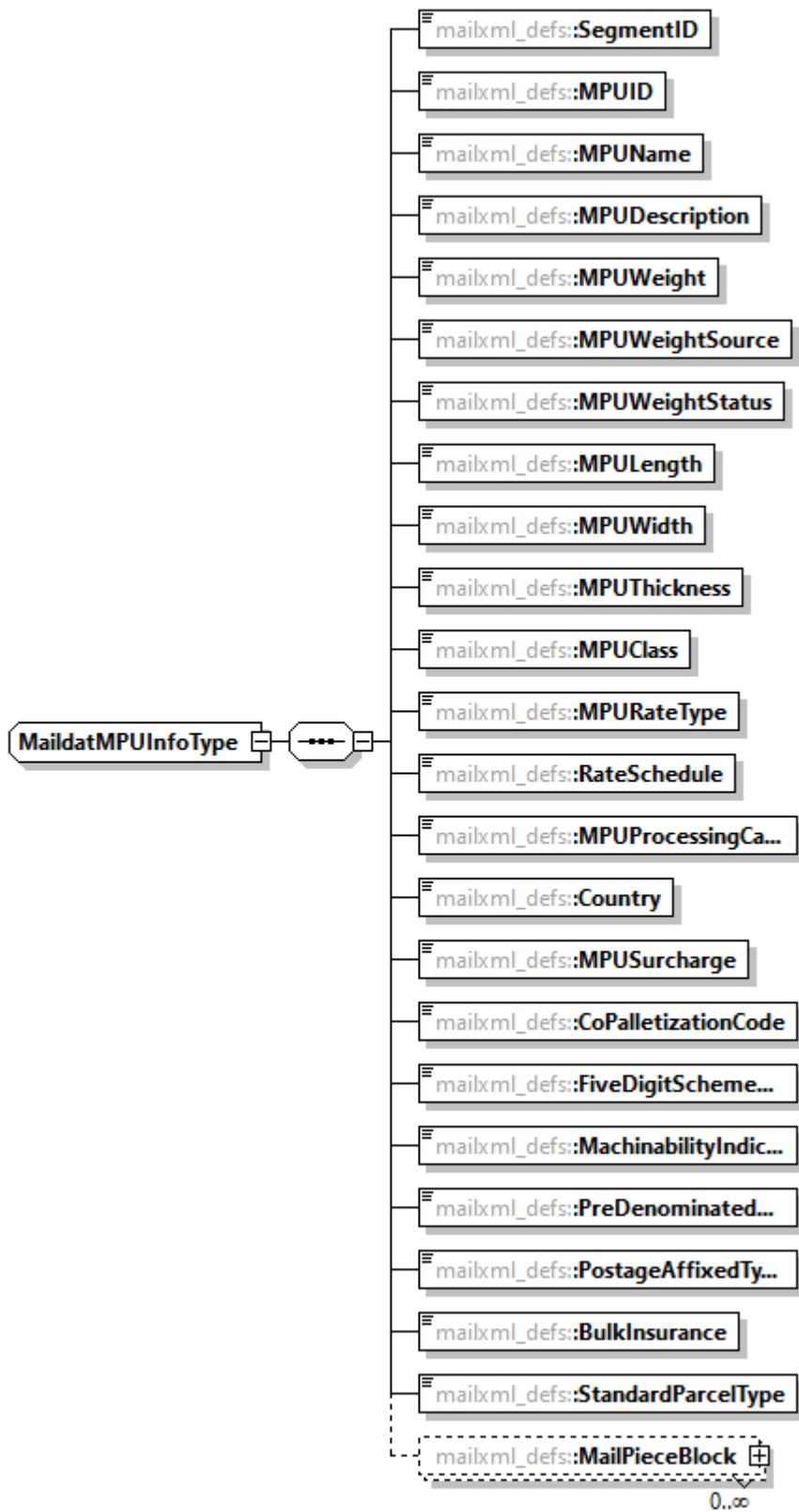
diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **maildatContainerIDType**

diagram	 <p>The diagram illustrates the structure of the maildatContainerIDType complex type. It features a central box labeled maildatContainerIDType with a small square icon on its right side. To the right of this box is a large container labeled attributes with a square icon. Inside the attributes container is a group labeled grp mailxml_defs:SiblingIdent... with a square icon. This group contains four dashed-line boxes, each representing an attribute: mailxml_defs:UserLicenseCode, mailxml_defs:MaildatJobID, mailxml_defs:ContainerID, and mailxml_defs:UniqueContainer.... Below the group, a text label reads: "Group of attributes identifying an original or sibling container".</p>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType MaildatMPUInfoType

diagram



namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
-----------	---

complexType **mailPieceldType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **MailXMLContainerIDType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **mailxmlDetailType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **measurementType**

diagram	<pre> sequenceDiagram measurementType --> seq1 seq1 --> choice choice --> Value[mailxml_defs::Value] choice --> RangeMin[mailxml_defs::RangeMin] choice --> RangeMax[mailxml_defs::RangeMax] </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **MIDType**

diagram	<pre> sequenceDiagram MIDType --> seq1 seq1 --> choice choice --> MID6[mailxml_defs::MID6] choice --> MID9[mailxml_defs::MID9] </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **MPSRBlockType**

diagram	<pre> sequenceDiagram MPSRBlockType --> seq1 seq1 --> choice choice --> ContainerID[mailxml_defs::ContainerID] ContainerID --> 1inf[1..∞] choice --> PSRBlock[mailxml_defs::PSRBlock] PSRBlock --> 1inf[1..∞] </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **MPSVisIncludedInScanRecFlagType**

diagram	<pre> sequenceDiagram MPSVisIncludedInScanRecFlagT... --> seq1 seq1 --> choice choice --> IMcb[mailxml_defs::IMcb] choice --> IMtb[mailxml_defs::IMtb] choice --> BundleID[mailxml_defs::BundleID] </pre>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

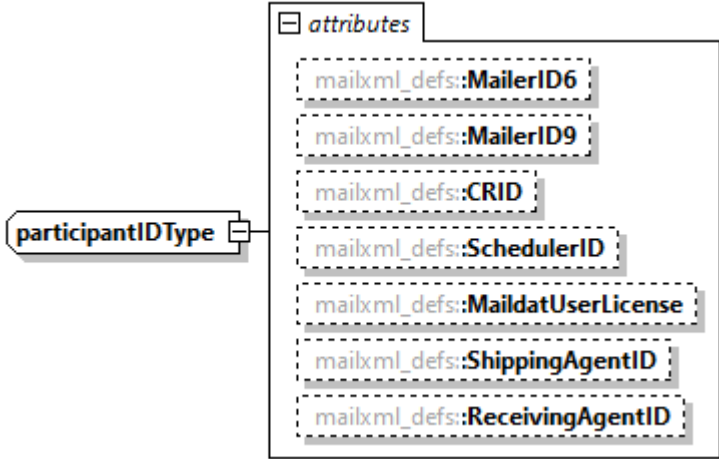
complexType **MPSVisResultOptionsType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

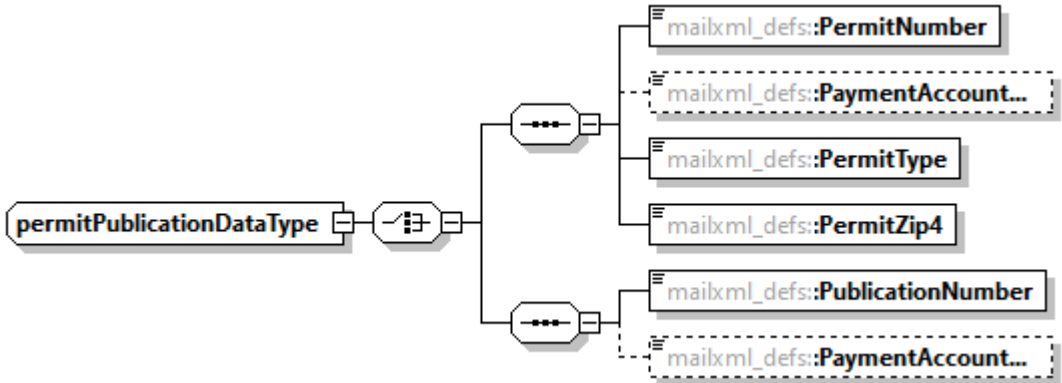
complexType **palletHandoffInfoType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **participantIDType**

diagram	 <p>The diagram shows a complex type named participantIDType. It has a list of attributes under the namespace <code>mailxml_defs:</code>. The attributes are: MailerID6, MailerID9, CRID, SchedulerID, MaildatUserLicense, ShippingAgentID, and ReceivingAgentID. Each attribute is represented by a dashed box with its name inside.</p>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **permitPublicationDataType**

diagram	 <p>The diagram shows a complex type named permitPublicationDataType. It contains a sequence of elements under the namespace <code>mailxml_defs:</code>. The elements are: PermitNumber, PaymentAccount..., PermitType, PermitZip4, PublicationNumber, and PaymentAccount.... Each element is represented by a dashed box with its name inside. The elements are connected by a sequence connector (a box with three dots).</p>
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **pickupApptBlockResponseType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

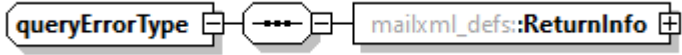
complexType **postageStatementSummaryType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

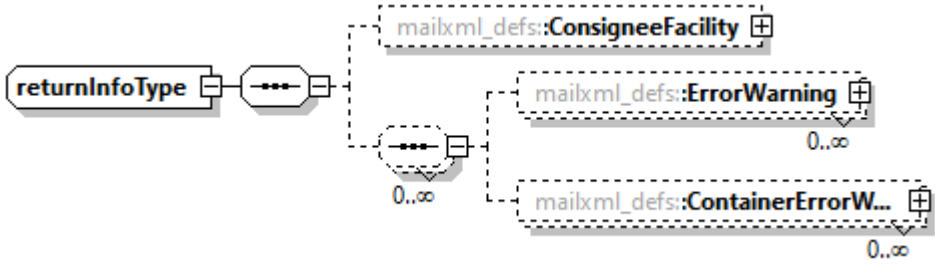
complexType **PSRBlockType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

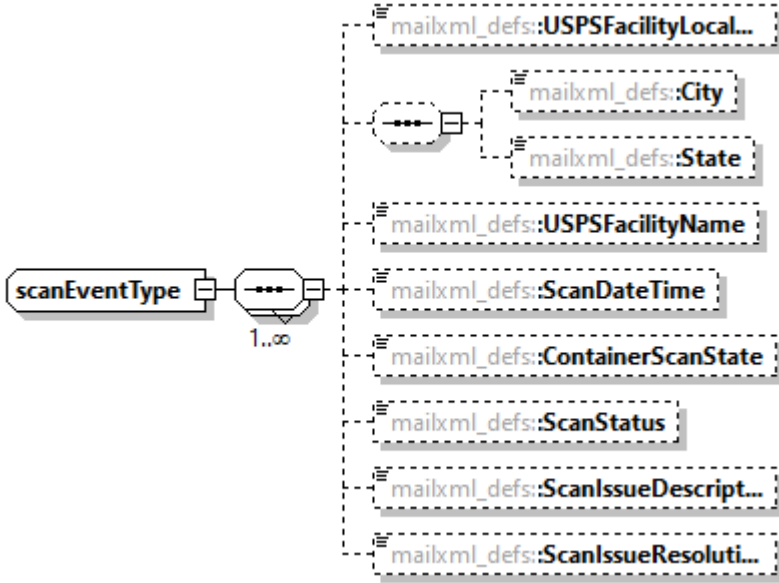
complexType **queryErrorType**

diagram	 The diagram shows the queryErrorType complex type. It consists of a queryErrorType element (rectangle with rounded ends) connected to a sequence container (octagon with three dots). This container is connected to a mailxml_defs:ReturnInfo element (rectangle with rounded ends and a plus sign in the top right corner).
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **returnInfoType**

diagram	 The diagram shows the returnInfoType complex type. It consists of a returnInfoType element (rectangle with rounded ends) connected to a sequence container (octagon with three dots). This container is connected to a choice container (octagon with three dots). The choice container has three branches: 1) mailxml_defs:ConsigneeFacility (rectangle with rounded ends and a plus sign). 2) mailxml_defs:ErrorWarning (rectangle with rounded ends and a plus sign), with a cardinality of 0..∞. 3) mailxml_defs:ContainerErrorW... (rectangle with rounded ends and a plus sign), with a cardinality of 0..∞.
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **scanEventType**

diagram	 The diagram shows the scanEventType complex type. It consists of a scanEventType element (rectangle with rounded ends) connected to a sequence container (octagon with three dots). This container is connected to a choice container (octagon with three dots). The choice container has several branches: 1) mailxml_defs:USPSFacilityLocal... (rectangle with rounded ends). 2) A sequence container (octagon with three dots) containing mailxml_defs:City and mailxml_defs:State (both rectangles with rounded ends). 3) mailxml_defs:USPSFacilityName (rectangle with rounded ends). 4) mailxml_defs:ScanDateTime (rectangle with rounded ends). 5) mailxml_defs:ContainerScanState (rectangle with rounded ends). 6) mailxml_defs:ScanStatus (rectangle with rounded ends). 7) mailxml_defs:ScanIssueDescript... (rectangle with rounded ends). 8) mailxml_defs:ScanIssueResoluti... (rectangle with rounded ends). The cardinality for the main choice container is 1..∞.
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **submittingSoftwareType**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs

complexType **zipCode**

diagram	
namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	extension of mailxml_base:ns09

simpleType **bundleScanTypeType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string

simpleType **containerDiscrepancyCategoryType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string

simpleType **containerScanStateType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string
annotation	documentation Scan data for Container Scan States

simpleType **containerStatusType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string

simpleType **countTypeType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string

simpleType **einductionDataSourceType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string

simpleType **fullServiceComplianceIndicatorType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	mailxml_base:yesNo

simpleType **MPSQueryType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string
annotation	documentation Mail Piece Scan Query Type

simpleType **MPSSStateType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string
annotation	documentation Scan States for Mail Piece Scan Data

simpleType **pieceScanEventTypeType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string

simpleType **problemCategoryType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string

simpleType **problemTypeType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string

simpleType **reasonCodeType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string

simpleType **retrieveDataByType**

namespace	http://delivery-tech.org/Specs/mailxml26.2/mailxml_defs
type	restriction of xs:string