API Specification

Royal Mail Group

Shipping API V1 (SOAP)

Technical User Guide

This API specification details the requirements for integrating with Shipping API V1 (SOAP). It specifically covers how the Shipping API V1 can be used by business customers to conduct shipping activity with Royal Mail and provides the technical information to build this integration. This specification must be used with the relevant accompanying specifications for customers wishing to interface their systems with Royal Mail services.

15th November 2015 Version 1.5

Contents

1 Document Control

1.1 Terms and Abbreviations

Table 1 – Terms and Abbreviations

Term	Meaning
Allocated	Shipment with a Service Type / Service / Service Format and shipment number but not printed
Base64	A standard binary-to-text encoding scheme that is used to represent binary data in an ASCII string format. Used to include binary data with an XML structure
DMO	Despatch Manager Online
HTTPS	Hypertext Transfer Protocol over SSL
IP	Internet Protocol
Local Collect	A Service Enhancement where delivery is made to an agreed alternate location
Manifested	Customer Collection Receipt has been created and Customer Collection Receipt has been printed
Nonce	User generated random string that is used as part of the password hashing process to ensure that messages cannot be replayed
OBA	Online Business Account
Printed	Shipment with Service Type / Service / Service Format and shipment number and the label(s) printed
SOAP	Simple Object Access Protocol
WSDL	Web Service Description Language
XSD	XSD is XML Schema Definition – this defines a specific template for the XML used by particular systems
XML	eXtensible Markup Language, a flexible standard for data inside a hierarchical structure of named data items

1.2 Version History

Table 2 – Document Version History

Version	Date	Notes
1.0	22/05/2014	Baselined with baselined WSDLs, XSD files and Service Matrix supporting files and feedback from Adrian Tharby
1.1	23/05/2014	Updated with new versions of WSDLs, XSD and Canonical Model files
1.2	14/07/2014	Update to add Local Collect as a Service Enhancement
1.21	18/07/2014	Updated following Chris Vaughan review
1.22	22/07/2014	Updated lists of Error Codes and Warning Codes
1.23	24/07/2014	Links to Royal Mail Services and SAPI Management User Guide

Version	Date	Notes
1.24	14/08/2014	Updated list of Error Codes and Warning Codes- Added Throttling Error E0010
1.3	05/09/2014	Update to add Special Delivery Saturday Guarantee as a Service Enhancement. Added content to descriptions on API usage.
1.4	25/11/2014	Update to remove Special Delivery Saturday Guarantee as a Service Enhancement. Document updated following peer review and following feedback. A number of updates in the API description where sample xml examples have been added.
1.5	15/11/2015	Document updated following introduction of API Management capability.

2 Overview

Royal Mail Shipping API exposes web services that allow account customers to create shipments, produce labels, and produce documentation; all the tasks required for them to ship items with Royal Mail. Built on industry standards, Shipping API provides a simple and low cost method for customers to integrate with Royal Mail, and allows them to get shipping quickly.

There are no costs to customers for using the Shipping API services, however customers' own development costs must be covered by the customer developing the solution. Royal Mail will not accept any responsibility for these development, implementation and testing costs.

Customers should address initial enquiries regarding development of systems for these purposes to their account handler.

3 Purpose

This document is to provide Royal Mail's customers with guidelines and detailed specifications for integrating with Royal Mail via the Shipping API web services. The document details:-

- The specification for the web service interfaces
- Description of errors the API can return
- Non-functional characteristics of the API including response times, service availability and security considerations

This document is primarily intended to be read by Developers and other technical roles involved with integrating customer systems' with the Shipping web service API. This document should be read in conjunction with the following artefacts which are available from the Shipping API V1 page on the Royal Mail Developer Portal:

- Shipping WSDL
- Shipping XSDs
- Shipping API Reference Data
- Shipping API FAQ
- Shipping API Sample Code
- Shipping API Sample Data

The web service operations included in the document are:

- createShipment
- updateShipment
- cancelShipment
- printLabel
- createManifest
- printManifest

For details of how to set up GUI user IDs and their viewing permissions, as well as creating reports, see the Shipping API Management User Guide, available from the Shipping API page on the Royal Mail Developer Portal. This set up is needed for both the Onboarding and Production Environments.

4 Introduction to Shipping API

Shipping API provides the functionality for customers to take a shipping transaction from creation to collection.

In simplest terms, the logical flow is as follows:

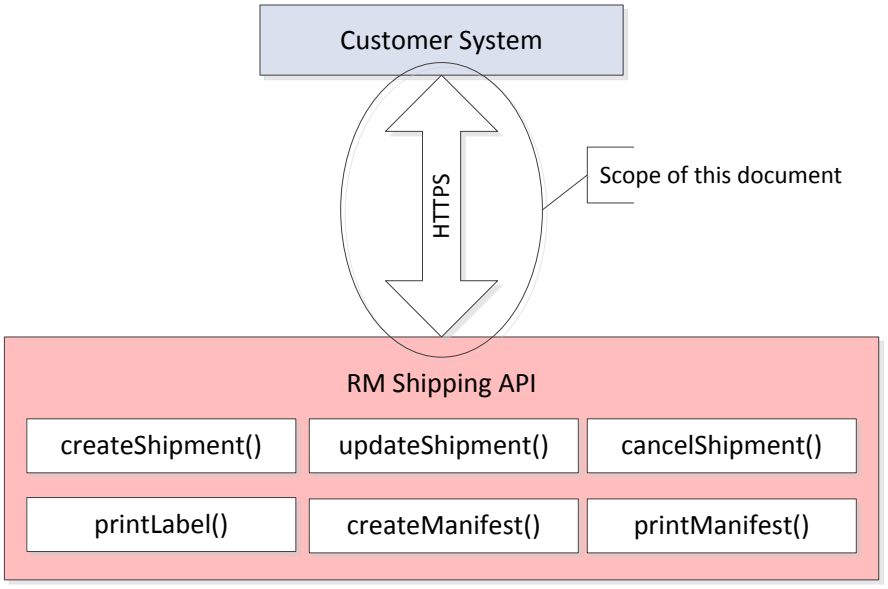
- Create Shipment – the details of an item are provided to Royal Mail, and a shipment is created on the system. The status of the shipment is 'Allocated'.
- Print Label – once a shipment has been created, the label for it can be printed. Once printed, the status of the shipment is updated to 'Printed'.
- Manifest Shipments – before items are collected, the customer must submit details of all the items to Royal Mail and print off the Customer Collection Receipt for the driver. The createManifest call submits the details for all shipments that are in the 'Printed' state to Royal Mail (those that are in the 'Allocated' state are ignored). The status of these shipments is then set to 'Manifested', and they can no longer be updated or cancelled.

Unless they have been granted an exemption, customer shipments will be subject to a clean sweep process. This is a process that runs at a specific time each night, and automatically manifests any 'Printed' shipments that have not already been manifested.

4.1 Interface Components

Please see Error! Reference source not found. below for a graphical representation of the interface between Royal Mail and customers for Shipping API. This document covers what information is to be exchanged, how this information is structured and the means by which it is transferred.

Figure 1 – Shipping API Interface



5 Integrating with the API

The high-level process associated with integrating with Shipping API V1 is represented in the diagram below and described in the sections which follow.

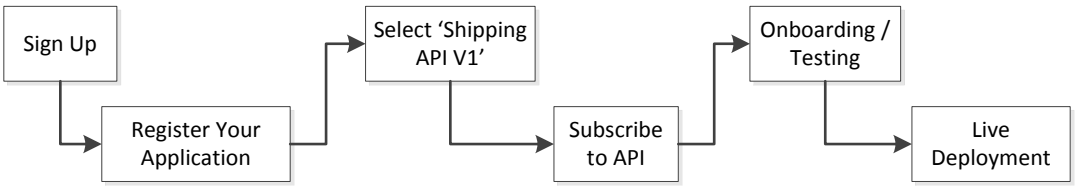


Figure 2 – Process for Integrating with the API

Access to the service is managed through RMG's API Management system.

New users of the system will need to:

1. Sign up for an account and accept the terms and conditions on the Royal Mail Developer Portal.
2. Register the customer side 'application' which will be calling the API. When the application is registered, it will be assigned a unique system-generated Client ID and Secret which is needed to securely access the API. It is important that these credentials are noted and securely stored.
3. Request to subscribe to the API. This will result in an e-mail being automatically generated and sent to the Royal Mail Customer Solutions team.
4. Once approved, testing can be performed against the API in a sandboxed onboarding environment that allows you to test the integration.
5. Once all required testing has completed in the onboarding environment, access to the Live production system will be provided at a mutually agreed date/time.

Existing users who already have an account with Royal Mail's API Management system will need to perform step 2 onwards if the application accessing the API is different to any currently registered applications. If the application accessing the API is already registered, existing customers will need to perform step 3 onwards.

5.1 Terms & Conditions

You must accept the Royal Mail Terms and Conditions when creating your customer account. These cover the ways in which the service may be used and any integration activities must abide by these.

Of particular note to developers:

- The onboarding environment may not be used for performance testing. This is a small scale system for functional testing only.
- Repeated reprints of labels or Customer Collection Receipts will be flagged to Royal Mail and may result in an investigation.

- Where specified, weights should be accurate. Discrepancies between reported and actual weights will be investigated by Royal Mail.
- All Royal Mail APIs impose a cap on the number of transactions per second for each customer. Excessive volumes of traffic within a short period will result in transactions being rejected.

The Customer Data will be held by us on our servers for a limited period of time (currently thirteen months) to enable users to run activity reports. Thereafter, we may delete or remove such Customer Data without further notice.

5.2 API Access

Both onboarding and live access to the API is obtained via the following URL:

https://api.royalmail.net/shipping/v1

Please note that the Client ID and Secret must be provided in the HTTP header of all API requests otherwise access to the API will be rejected and a HTTP 401 (Unauthorised) will be returned. The Client ID and Secret are obtained by registering an application on the RMG API Management site.

You must complete all required test activities in the onboarding environment prior to being permitted access to the live environment by the Royal Mail Customer Solutions Team. The onboarding test environment is available 24x7, has the same functionality as live (though with a reduced capacity) and allows you to test your integration without data being passed through to the Royal Mail operational and billing systems and without incurring any charges against your account.

You will be provided with a contact in Royal Mail who will take you through the onboarding process and provide you with the required security credentials to access the API. Once you have successfully demonstrated that your system works with ours, and that you can produce labels to the required level of quality, you will be granted access to the live system and can begin shipping items

5.3 Live Deployment

Once you have completed all required testing in the onboarding environment you will be provided with access to the live production system.

If new products or services are added to your account, you may be asked to demonstrate that you have these working correctly in the onboarding environment before you are allowed to use them on the live system.

5.4 API Versioning

Royal Mail is continuously working to improve its technology, and as part of this process updates to the services provided may on occasion necessitate a new WSDL version. Royal Mail will look to maintain three versions of the WSDL; as new versions are introduced, previous versions move down the stack until they are ultimately removed completely:

- Latest version
- Previous version

- Deprecated version

Customers will always be encouraged to integrate against the latest version as this will give them the longest stable period without the need to change, but if they have already begun integration activities when a new version is released then they will be able to integrate against the previous version. Customers should not integrate against the deprecated version.

6 Shipping Services

6.1 Business Services

Shipping API is a service offered via RM to request for the creation / update / cancellation of a shipment, printing of a label and creation / printing of a manifest. The table that follows provides an overview of the business services that are supported by this interface.

Table 1 – Business Services

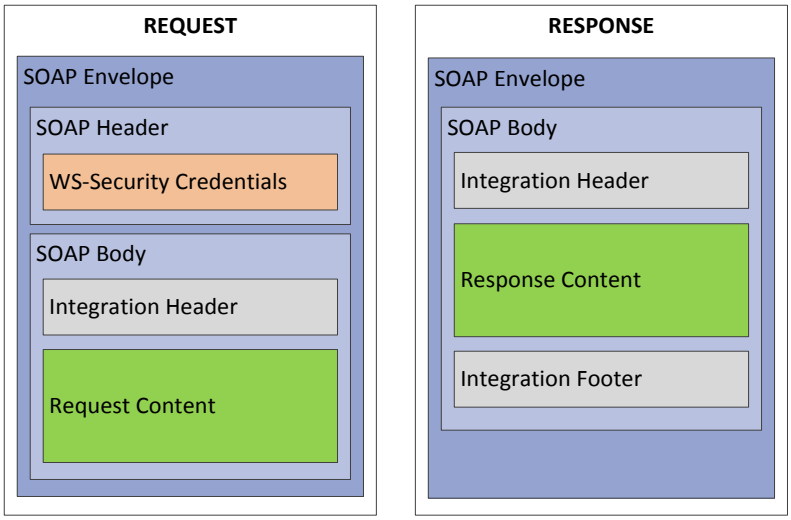
Business Service	Description	Request Message	Response Message
Create Shipment	Creates a shipment on the system	createShipmentRequest	createShipmentResponse
Update Shipment	Updates the details of a shipment that has been created but not manifested	updateShipmentRequest	updateShipmentResponse
Cancel Shipment	Cancels a shipment that has been created but not manifested	cancelShipmentRequest	cancelShipmentResponse
Print Label	Prints a label for a shipment that has been created	printLabelRequest	printLabelResponse
Create Manifest	Manifests current shipments	createManifestRequest	createManifestResponse
Print Manifest	Provides a printable manifest in PDF format	printManifestRequest	printManifestResponse

All of the above services follow a synchronous request / response conversation style.

7 Message Structure

Please see **Error! Reference source not found.** for an overview of the high-level request / response message structure of the API.

Figure 3 – API Structure



Each of the Shipping API operations (in section 6.1) is defined as a SOAP operation, with separate request / response message parts. Technical details are documented in the WSDL and XSD schemas which accompany this document.

Within the SOAP body tags, each request or response message is supplemented with an integrationHeader element and is described in Section 8.4. Response messages may contain an integrationFooter element that contains details of any errors or warnings and is described in Section 8.5.

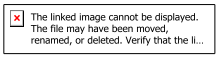
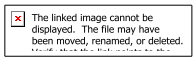
8 Message Definition and Schemas

8.1 Notation

The API elements described in this section are constructed using Royal Mail's Enterprise Canonical Data Model.

These standard elements contain optional fields not used by the Shipping API web service – for example the standard Address element contains optional elements such as addressUsageType, buildingName, and buildingNumber. If an element is not listed below, it is not used by the Shipping API web service and data should not be placed in those elements. Any elements not used in the Shipping API implementation are shaded in grey and indicated as such in the supporting text. Elements described in more detail later are shaded in purple.

The schema structures are described in XMLSpy notation with graphical representation meanings explained in the table below.

Graphical Representation	Meaning
	A solid line around an element indicates it is a mandatory field that will/must always be present.
	A dashed line around an element indicates the field is optional and may or may not be present. An optional element is one that indicates **the data and enclosing XML tags** may or may not be present in either a request or response.
	The *sequence* identifier represents an ordered set of elements.
	The *choice* identifier represents a selection of elements.

In the context of the tables defined in this section:

- All data types are defined in the Data Types schema (see section **Error! Reference source not found.**).
- All 'Occurs' values signify whether each data entity is mandatory or optional in the context of Shipping API. Although an element may be defined as optional in the Canonical Data Model, it may still be considered as being mandatory for a Shipping API request. Particular conditions under which an element needs to be populated are defined in the 'Description' column.

8.2 WSDL & Schemas

This section includes the full definition on the Shipping Web Service – documented through supporting WSDL and XSD files. These are provided in a separate file.

Table 2 – Shipping API WSDL and Schemas

WSDL/ XML Schema file	Description
ShippingAPI_V1.0.wsdl	WSDL describing the functionality offered by Royal Mail Shipping API
ShippingAPI_V1.0.xsd	Schema defining the data structures used by the Shipping API
oasis-200401-wss-wssecurity-secext-1.0.xsd	OASIS schema defining the standards for OASIS WS Security
oasis-200401-wss-wssecurity-utility-1.0.xsd	OASIS schema that supplements the OASIS WS Security schema
xml.xsd	Schema for the XML language, used to support the Oasis schema files
xmldsig-core-schema.xsd	XML Signature schema used to support the Oasis WS Security standard
CommonClassesV2_2.xsd	Royal Mail (RMG) schema which defines common objects such as "address" which are complex types with a defined structure based on RMG defined simple data types as well as reference data types.
DatatypesV2_2.xsd	RMG defined simple types that are commonly required such as "date" and "name". This schema is the foundation for the other schema files.
CommonIntegrationSchemaV1_9.xsd	RMG schema which defines a common header definition to be used for request and response messages.
ReferenceDataV2_2.xsd	RMG schema which defines a set of types for common reference data such as address fields that are defined using the RMG simple data types.

8.3 HTTP Header Information

8.3.1 Description

The purpose of the HTTP header is to support security and logging functionally within the Royal Mail systems and it is mandatory that it is provided in the request message.

8.3.2 Request Message

All service requests to this API will be authorised in accordance with the Client ID and Secret passed in the HTTP headers. Please see table below for the elements which need to be populated in the HTTP header.

Table 3 – HTTP Header Information in the API Request

Parameter	Optional	Description
X-IBM-Client-Id	No	Similar to a client username. Required to access the API.
X-IBM-Client-Secret	No	Similar to a client password. Required to access the API.

8.3.3 Example Data

Example request data for the HTTP Header:

Table 4 – Example HTTP Header Information for API Request

Parameter	Value
X-IBM-Client-Id	f0e4f151-2041-4df2-b31d
X-IBM-Client-Secret	kT0lB2dK0wF6mK0rD8sD7oE7vP2mG7l

8.4 integrationHeader element

8.4.1 Description

The integrationHeader element is present in all request and response messages, and is used to support security and audit functionality within Royal Mail systems.

8.4.2 Request Message

All elements are relative to the integrationHeader element in the request message.

integrationHeader element

Figure 4 – integrationHeader Structure

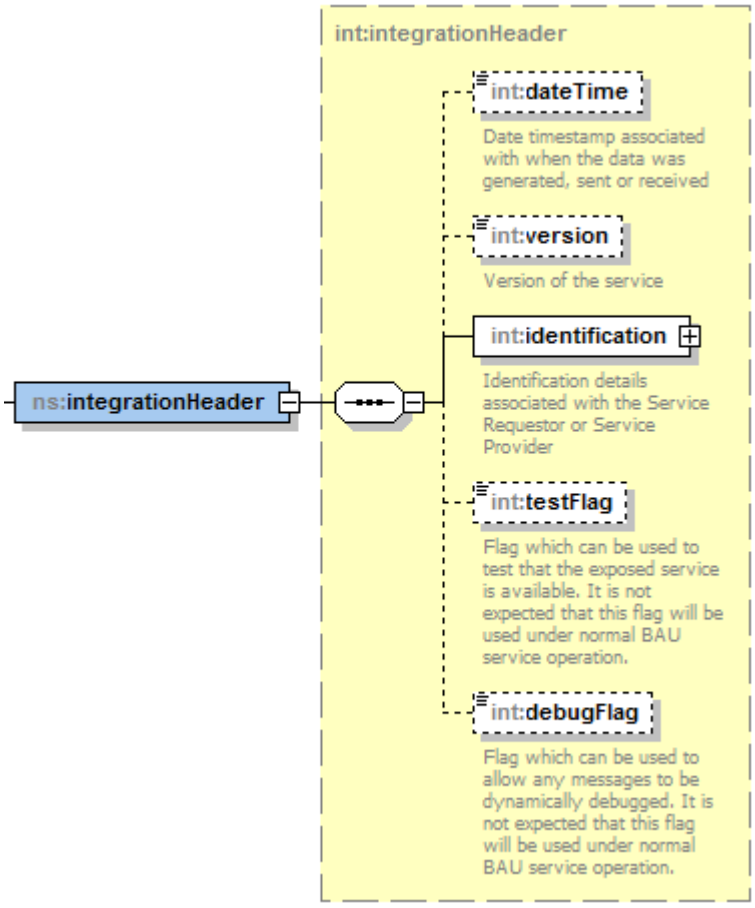


Table 5 – integrationHeader Element (Request)

Element	Occurs	Data Type (Refer section Error! Reference source not found.)	Description
dateTime	0-1	dateTime	This should be populated with the date timestamp when the message was generated.
version	0-1	decimal	The version of the API currently being used (currently 1.0).
identification	1-1	element	Identification element used to hold the identity and transaction details associated with the customer. See identification table below for structure.
testFlag	0-1	boolean	Not used in the Shipping API implementation.

Element	Occurs	Data Type (Refer section Error! Reference source not found.)	Description
debugFlag	0-1	boolean	Not used in the Shipping API implementation.

identification element

Figure 5 – identification Structure

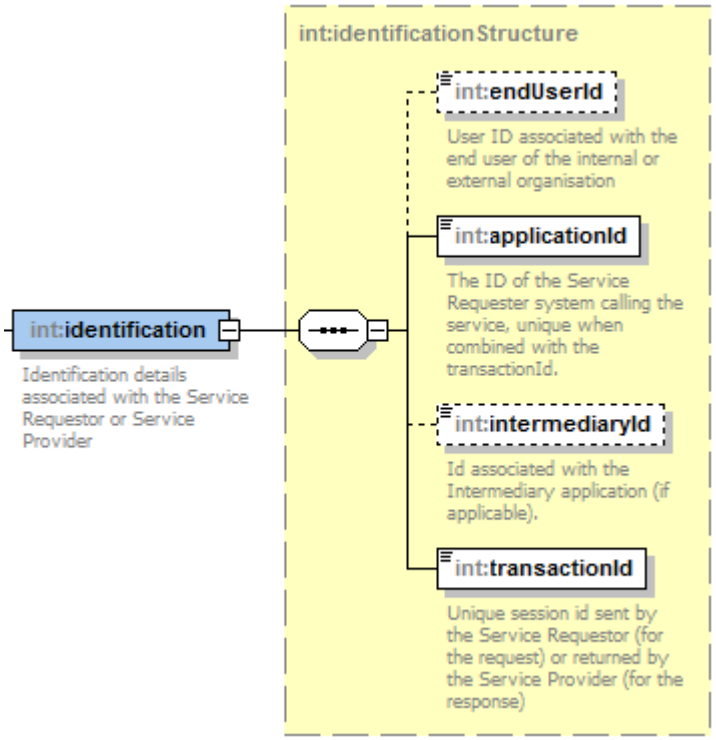


Table 6 – integrationHeader identification Element (Request)

Element	Occurs	Data Type	Description
enduserID	0-1	identifier	Not used in the Shipping API implementation.
applicationId	1-1	identifier	This is the ten digit Customer Account Number allocated by Royal Mail
intermediaryId	0-1	identifier	Not used in the Shipping API implementation.
transactionId	1-1	identifier	This is a unique number used to identify the transaction as provided by the customer system. Any value can be provided in this field but must contain only the characters 'a-z', 'A-Z', '0-9', '/' and '-'. It allows the consuming application to correlate the response message to its request.

8.4.3 Response Message

All elements are relative to the integrationHeader element in the response message.

integrationHeader element

Table 7 – integrationHeader Element (Response)

Element	Occurs	Data Type	Description
dateTime	0-1	dateTime	This is always returned and contains the same value provided in the request.

Element	Occurs	Data Type	Description
version	0-1	decimal	This is always returned and contains the same value provided in the request.
identification	1-1	element	This is always returned and contains the same values provided in the request.
testFlag	0-1	boolean	Not used in the Shipping API implementation.
debugFlag	0-1	boolean	Not used in the Shipping API implementation.

identification element

Table 8 – intgrationHeader identification Element (Response)

Element	Occurs	Data Type	Description
enduserID	0-1	identifier	Not used in the Shipping API implementation.
applicationId	1-1	identifier	This is always returned with the same ten digit Customer Account Number provided in the request.
intermediaryId	0-1	identifier	Not used in the Shipping API implementation.
transactionId	1-1	identifier	This is always returned with the same unique transaction number provided in the request

8.4.4 Example Data

Full XML examples of SOAP requests and responses are provided in Section 0 at the end of this document. This section provides a simplified extract to illustrate the integrationHeader only.

```
<v1:integrationHeader>
  <v11:dateTime>2013-12-12T15:29:42</v11:dateTime>
  <v11:version>1.0</v11:version>
  <v11:identification>
    <v11:applicationId>111111113</v11:applicationId>
    <v11:transactionId>420642961</v11:transactionId>
  </v11:identification>
</v1:integrationHeader>
```

8.5 integrationFooter element

8.5.1 Description

The purpose of this element is to return any business error and warning messages back to the customer. The integrationFooter element will only be present in a response message if there are any business errors or warnings associated with processing a web service request.

Please note that all technical exceptions will result in a SOAP fault being generated. We can provide details of all SOAP fault conditions which can be returned.

integrationFooter

Figure 6 – integrationFooter Structure

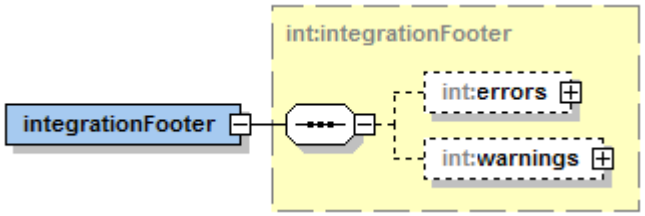


Table 9 – integrationFooter Element

Element	Occurs	Data Type	Description
errors	0-1	element	See table below for structure. This will contain details of any errors encountered in processing a web service request. We can provide a full list of all errors which can be returned.
warnings	0-1	element	See table below for structure. This will contain details of any warnings encountered in processing a web service request. We can provide a full list of all warnings which can be returned.

errors

Figure 7 – integrationFooter errors Structure

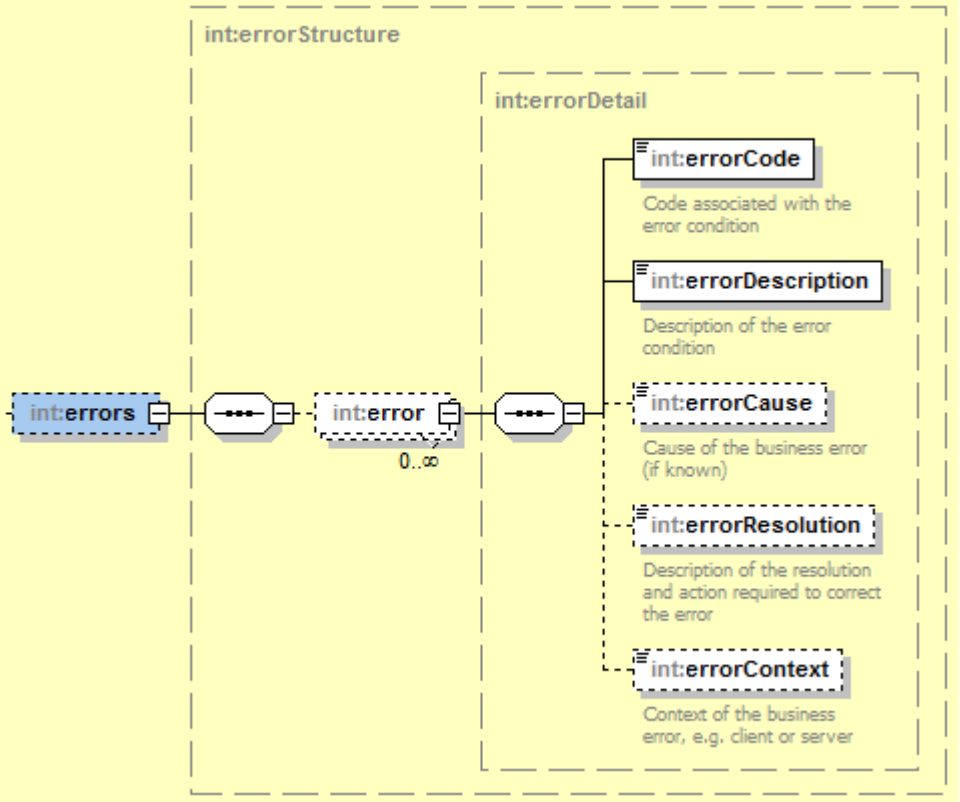


Table 10 – integrationFooter errors Element

Element	Occurs	Data Type	Description
error	0-∞	element	Element containing from zero to many errors.
error/errorCode	1-1	identifier	This is the code for the error message.
error/errorDescription	1-1	description	This is the description associated with the error code.
error/errorCause	0-1	description	This is the cause of the error (if known).
error/errorResolution	0-1	description	This is the description of the resolution and action required to correct the error (if known).
error/errorContext	0-1	description	This is the context of the error, e.g. client or server.

warnings

Figure 8 – integrationFooter warnings Structure

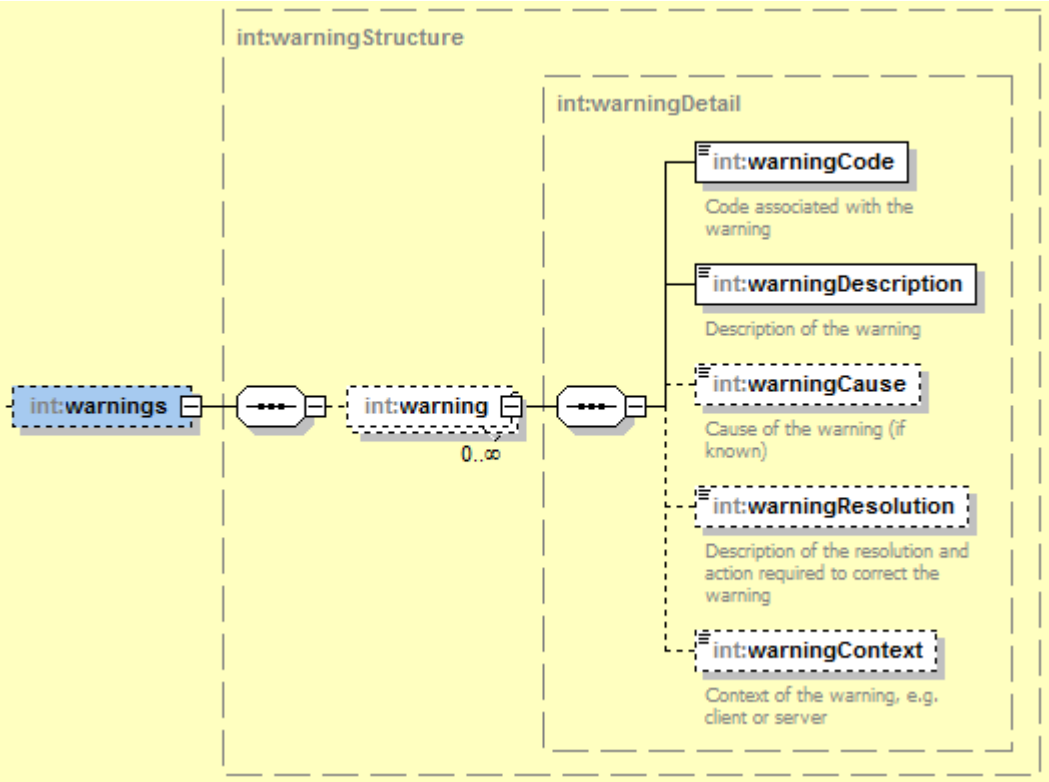


Table 11 – integrationFooter warnings Element

Element	Occurs	Data Type	Description
warning	0-∞	element	Element containing from zero to many warnings.
warning/warningCode	1-1	identifier	This is the code for the warning message.
warning/warningDescription	1-1	description	This is the description associated with the warning code
warning/warningCause	0-1	description	This is the cause of the warning if known.
warning/warningResolution	0-1	description	This is the description of the resolution and action required to correct the warning if known.
warning/warningContext	0-1	description	This is the context of the warning, e.g. client or server.

8.5.2 Example Data

Full XML examples of SOAP requests and responses are provided in Section 0 at the end of this document. This section provides a simplified example to illustrate the integrationFooter only.

```
<ns:integrationFooter>
    <int:errors>
        <int:error>
            <int:errorCode>E1084</int:errorCode>
            <int:errorDescription>shipmentType is a required field</int:errorDescription>
        </int:error>
    </int:errors>
</ns:integrationFooter>
```

8.6 Security Credentials

8.6.1 Description

Each request message to the Shipping API web services must include valid security credentials in order to be accepted by the Royal Mail system. These comply with the OASIS WS security standards.

The credentials are encoded as defined in the OASIS standard *Web Services Security Username Token Profile version 1.0* and comprise:

- the username of the API user account,
- the SHA-1 hash of the password of the API user account.
- the Created timestamp for the request,
- and a Nonce.

The credentials are carried in a wsse:Security element included in the SOAP Header element of the request.

Sample code is provided on the 'Shipping API V1 (SOAP)' page on the Royal Mail Developer Portal to accelerate customer integration with Shipping API. The sample code assists with the population of the WS-Security element in the SOAP request message. The username and password required to call the API are separately supplied by Royal Mail as part of the customer onboarding process.

8.6.2 wsse:Security Element

All elements are relative to the wsse:Security element in the SOAP Header of the request message. Please note that the 'Occurs' column below is populated in the context of the Shipping API implementation (as opposed to the OASIS security standard).

Table 12 – wsse:Security Element

Element	Occurs	Description
Username	1-1	API username which will be supplied by RM
Password	1-1	Password Digest constructed as defined in ws-security Username Token Profile with the SHA-1 hash of the plain text password used as the password information. The plain text password will be provided to you by RM. See section below for details on how to construct this value.
Nonce	1-1	Random number. Although optional in the ws-security schema, this element must be provided by the customer for Shipping API web services.
Created	1-1	Date and time. Although optional in the ws-security schema, this element must be provided by the customer for Shipping API web service requests.

Please note that none of the optional attributes in the ws-security header need to be populated. If these are populated they will be ignored by the RM shipping system.

Nonce

1. A Nonce is a random number that the sender generates to include in each usernameToken.
2. The Shipping API web service maintains a cache of Nonce values for a period of 5 minutes. Upon receipt of the message, a check will be made to ensure the Nonce

has not already been used (i.e. it is not in the cache); messages containing Nonces that have already been used will be rejected.
3. For Shipping API the Nonce should be a 16 byte (128 bit) value.
4. The Nonce is presented as a Base64 encoded value in the wsse:Security element. Note: when used in creating the Password Digest, the Nonce must be used in unencoded form and not in Base64 form.

Created

1. Created Timestamp is the date and time the request message was created by the customer. The string format of the timestamp should be constructed as defined in the WS-Security standards: that is YYY-MM-DDThh:mm:ssZ.
2. Messages that contain a Created timestamp older than five minutes will be rejected. Customers are recommended to synchronise their systems with a common time source to help prevent this error condition from occurring.

Password

The OASIS Usertoken profile defines and describes the formula that computes the unique Password_Digest string submitted in the XML. For Shipping API the password used in the below formula is the base 64 encoding of the SHA-1 hash of the plain text password.
This password value is then concatenated to generate the password digest with the Nonce and Created Time. It will then be rehashed using SHA-1.

.

The formula to use to construct the Password_Digest value is

Password_Digest = Base64(SHA-1(Nonce + Created + SHA-1(Password)))

Note the **+** symbol in the above algorithm represents a string concatenation of the three strings: Nonce from xml request, Created from xml request and the SHA-1 digest of the Password.

When processing a request, Shipping API will:

- Check the Created timestamp is within five minutes of current time
- Check the Nonce has not been seen within the last five minutes
- Construct a Password_Digest using the Nonce and Created timestamp from the request message and the SHA-1 hash of API user account's password from its internal database.
- Compare the Password_Digest created with the one in the request message. If they match, authorisation will pass, otherwise it will fail and an authorization failure SOAP fault will be returned (see section 9).

8.6.3 Example Security data

Full XML examples of SOAP requests and responses are provided in Section 13 at the end of this document. This section provides a simplified example to illustrate the ws-security header only.

```
<soapenv:Header>
```

```
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
          <wsse:Username>ARCF5693CUSAPI</wsse:Username>
          <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest">bCjjYGOOpPr69p/dqqfIDMjGmKk=</wsse:Password>
          <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">xWstjXG0iUxbv3NH/fX+kw==</wsse:Nonce>
          <wsu:Created>2014-08-16T15:29:42</wsu:Created>
        </wsse:UsernameToken>
      </wsse:Security>
    </soapenv:Header>
```

8.6.4 References

For further details around the key concepts of WS-Security please see the following articles:

Microsoft Technet Article on WS-Security:
http://msdn.microsoft.com/en-gb/library/ms977327.aspx

8.7 createShipment Operation

8.7.1 Description

The createShipment operation either results in a shipment being created (based on the information provided in the request) or an appropriate error / warning message being returned to the customer.

8.7.2 createShipment Request Message

To invoke the createShipment operation, the customer shipping system constructs a SOAP request message described in Section 7.

The createShipmentRequest element is contained in the SOAP Body and contains the following:

- An integrationHeader element (see section 8.4)
- A requestedShipment element

createShipmentRequest element

Figure 9 – createShipmentRequest Structure

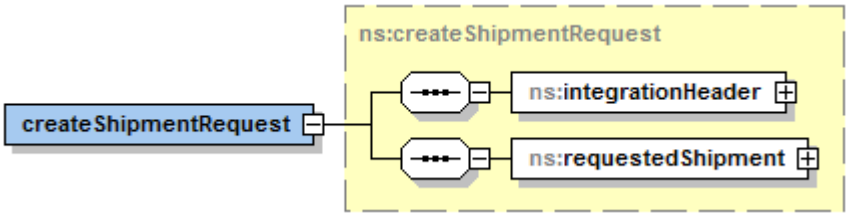


Table 13 – createShipmentRequest Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.4
requestedShipment	N/A	1-1	element	Container for the requested shipment details. See section below for more information. For allowable character set see section 12.1

requestedShipment element

Figure 10 – requestedShipment Structure

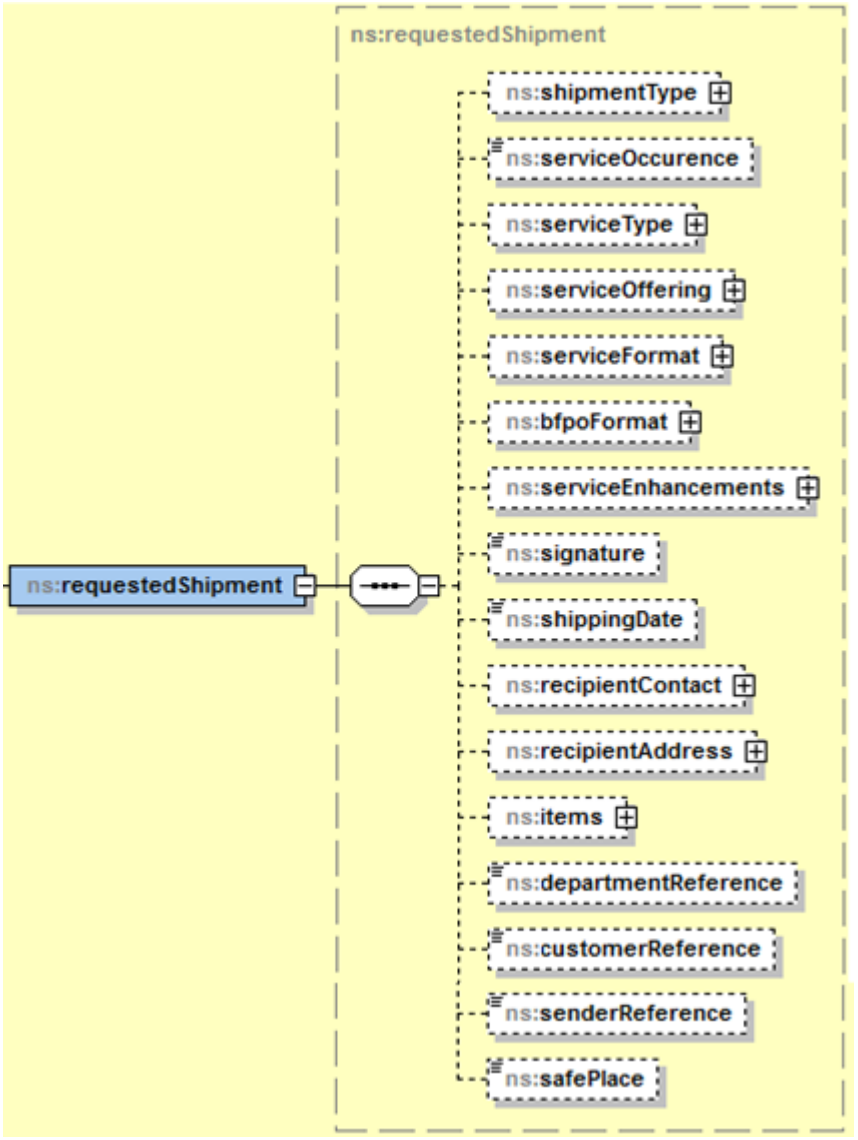


Table 14 – requestedShipment Element

Element	Max Length	Occurs	Data Type	Description
shipmentType	N/A	0-1	element	Container for shipmentType code.
shipmentType/code	8	0-1	identifier	**Mandatory**. Specifies whether the shipment being created is a standard delivery service or a returns service. Accepted values are '**Delivery**' and '**Return**'.

Element	Max Length	Occurs	Data Type	Description
serviceOccurence	2	0-1	ordinal	Part of the customer's contract identifier. In conjunction with the Service Offering it identifies an agreement line on the customer's account. If only one Service Reference exists then this is not required. No leading zero is required.
serviceType	N/A	0-1	element	**Mandatory**. Container for serviceType code
serviceType/code	4	0-1	identifier	**Mandatory**. The system Service Type of the shipment. We can provide a list of permissible values.
serviceOffering	N/A	0-1	element	**Mandatory**. Container for serviceOfferingCode
serviceOffering/ serviceOfferingCode/	N/A	1-1	element	**Mandatory**. Container for serviceOffering code value
../serviceOfferingCode/code	3	0-1	identifier	**Mandatory.** The Service Offering code for the mail item ordered. We can provide a list of permissible values. Please note that this field is case sensitive.
serviceFormat	N/A	0-1	element	Container for serviceFormatCode
serviceFormat/ serviceFormatCode	N/A	1-1	element	Container for serviceFormat code value
../serviceFormatCode/code	4	0-1	identifier	The Service Format code for the shipment. We can provide a list of permissible values. Note that this field is case sensitive.
bfpoFormat	N/A	0-1	element	Container for bFPOFormatCode
bfpoFormat/bFPOFormatCode	N/A	1-1	element	Container for bFPOFormat code value
../bFPOFormatCode/code	4	0-1	identifier	For HM Forces (BFPO) Service Type only when the Service Format is not International Flat, International Letter or International Packet. We can provide a list of allowable values.
serviceEnhancements	N/A	0-1	element	This is a complex type and is detailed below
signature	1	0-1	boolean	For RM Tracked items only, this element specifies whether a signature is required on delivery. If this element is not included then it defaults to false.
shippingDate	10	0-1	Date	This is the date that the item will be physically sent (in the format YYYY-MM-DD). This may be up to 28 days in the future. Please note that for returns a Shipping date must be provided.
recipientContact	N/A	0-1	element	**Mandatory**. This is a complex type and is detailed below
recipientAddress	N/A	0-1	element	**Mandatory**. This is a complex type and is detailed below
items	N/A	0-1	element	**Mandatory**. This is a complex type and is detailed below
departmentReference	10	0-1	identifier	This is the department reference code that customers can define in OBA. This is visible in the system departmental references GUI.
customerReference	12	0-1	identifier	This field allows customers to supply a reference that applies to multiple shipments and is included to mirror the functionality offered by the Customer Reference field in OBA, whereby a reference can be associated to a group of items. For references that apply to a single shipment, the senderReference field should be used. **Warning: Misuse of this field may result in incorrect billing.**

Element	Max Length	Occurs	Data Type	Description
senderReference	20	0-1	identifier	This field allows the user to supply their own reference number. Where supported (e.g. Tracked Returns) this number will appear on the label.
safePlace	30	0-1	comment	For Tracked non-signature service offerings only; this field allows a string that gives details of the recipient's designated safeplace (e.g. "inside the porch").

serviceEnhancements element

Figure 11 – serviceEnhancements Structure

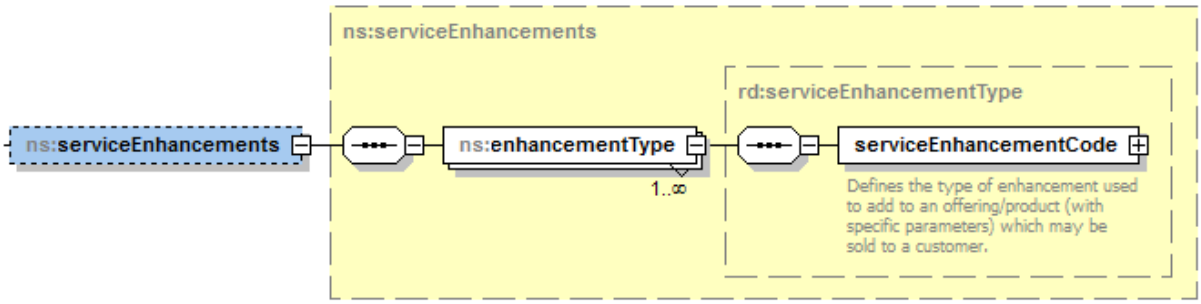
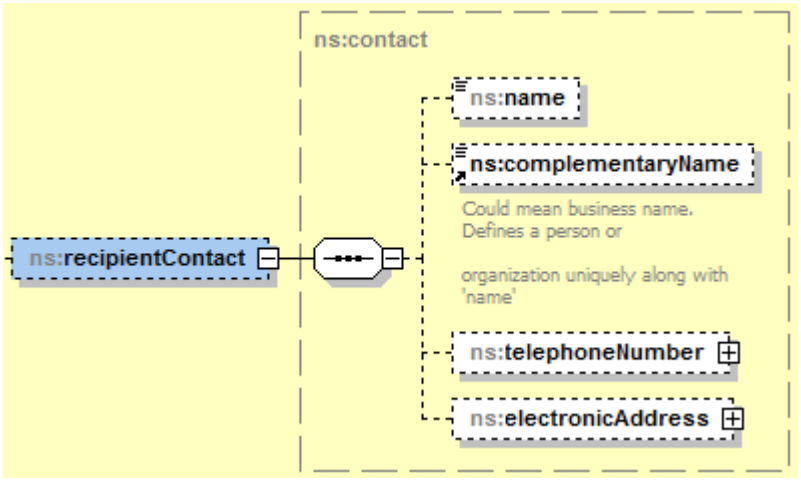


Table 15 – serviceEnhancements Element

Element	Max Length	Occurs	Data Type	Description
enhancementType	N/A	1-∞	element	Container for serviceEnhancementCode
enhancementType/ serviceEnhancementCode	N/A	1-1	element	Container for service enhancement code
../serviceEnhancementCode /code	4	0-1	identifier	If serviceEnhancement is included then enhancementType is Mandatory. There can never be more than one enhancementType specified from each Service Enhancement Group. Note that this field is case sensitive. We can provide details of enhancement types.

recipientContact element

Figure 12 – recipientContact Structure



Please note that all elements under the 'telephoneNumber' and 'electronic Address' elements and which are not listed in the table below, do not need to be populated.

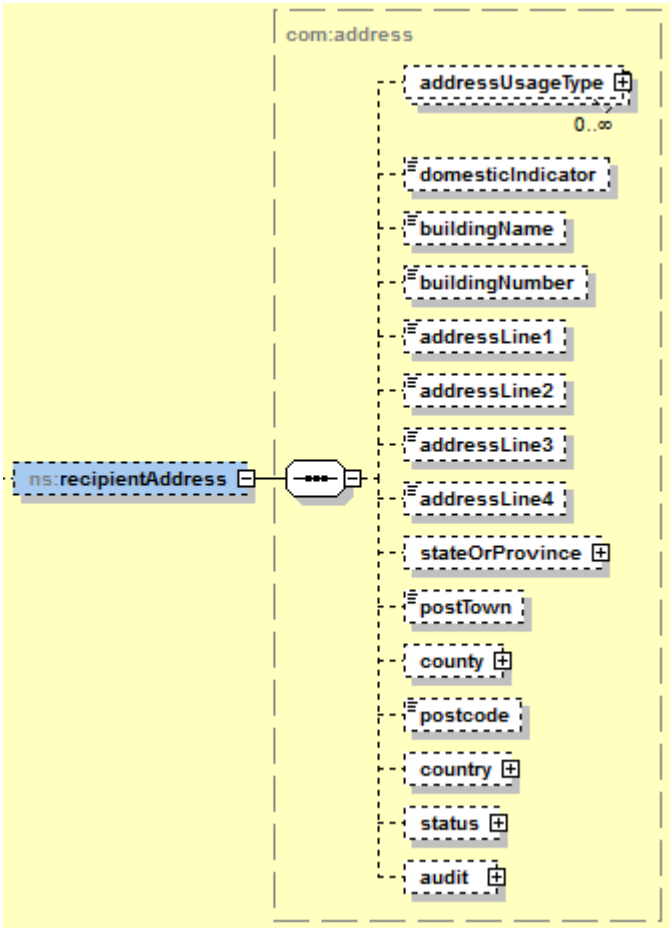
Table 16 – recipientContact Element

Element	Max Length	Occurs	Data Type	Description
name	80	0-1	longName	**Mandatory**. Contact name
complementaryName	64	0-1	longName	Business name
telephoneNumber	N/A	0-1	element	Container for telephone number details
telephoneNumber/ telephoneNumber	12	0-1	integer	UK mobile phone number. Required if a SMS service enhancement is selected.
electronicAddress	N/A	0-1	element	Container for electronic address details
electronicAddress/ electronicAddress	60	0-1	description	Email address to be used for notifications. Required if Email enhancement is selected

Note: If you specify more than 27 characters in name or complimentary name fields and generate a PDF label, these fields will be truncated to 27 characters in order to fit onto the label. It is therefore recommended that customers enter values within the 27 character display limit.

recipientAddress element

Figure 13 – recipientAddress Structure



Please note that elements which are not listed in the table below do not need to be populated.

Table 17 – recipientAddress Element

Element	Max Length	Occurs	Data Type	Description
addressLine1	80 (27)	0-1	description	**Mandatory**. First line of the address
addressLine2	80 (27)	0-1	description	Second line of the address
addressLine3	80 (27)	0-1	description	Third line of the address
postTown	40	0-1	name	**Mandatory**. Town or City
postcode	15	1-1	identifier	**Mandatory** for UK addresses. If the Shipment Type is "Return" then this must match the postcode of the registered return address.
country	N/A	0-1	element	Container for country details
country/countryCode	N/A	0-1	element	Container for country code details
../countryCode/code	2	1-1	identifier	The relevant country code for the address. A list of allowable values is available on request. For domestic services this must be GB. Note that this field is case sensitive.

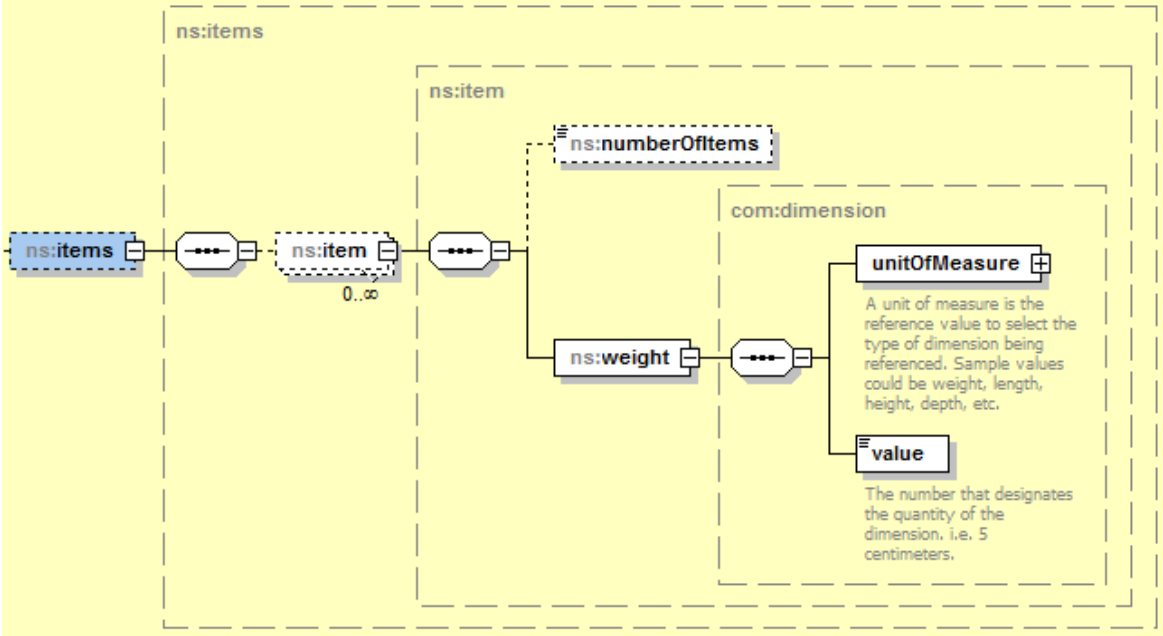
Note: If you specify more than 27 characters in addressLine1, addressLine2 or

addressLine3 fields and generate a PDF label, these fields will be truncated to 27 characters in order to fit onto the label. It is therefore recommended that customers enter values within the 27 character limit.

items element

The items element can contain multiple item elements. Each item element is constructed as follows:

Figure 14 – items Structure



Please note that elements which are not listed in the table below do not need to be populated.

Table 18 – items Element

Element	Max Length	Occurs	Data Type	Description
item	N/A	0- ∞	element	Container for item details
item/numberOfItems	2	0-1	cardinal	Number of items for the associated weight
item/weight	N/A	1-1	element	Container for item weight details
../weight/value	5	1-1	float	Weight in grams of each item (no decimal places). If the service has a weight band, for example Special Delivery, then the upper band will be used. For example, 150 grams will use the 100 to 200 grams band. Tracked services, for example, do not have a band and therefore take the actual weight.
../weight/unitofMeasure	N/A	1-1	element	Container for unit of measure
../../unitofMeasure/ unitofMeasureCode	N/A	1-1	element	Container for unit of measure code
../../unitofMeasure/ unitofMeasureCode/code	N/A	1-1	identifier	Must be 'g' for grams

8.7.3 CreateShipment Response Message

The response for a requested createShipment operation is constructed as a SOAP response message as described in Section 7.

The createShipmentResponse element is contained in the SOAPs Body element and contains the following:

- An integrationHeader element (see section 8.4)
- An optional completedShipmentInfo element (see table below) which will be populated on completion of having successfully fulfilled a createShipment request. The completedShipmentInfo element will not be present in the response if an error occurred in the processing of the request – the error details will be included in the integrationFooter element.
- An optional integrationFooter element (see section 8.5)

createShipmentResponse element

Figure 15 – createShipmentResponse Structure

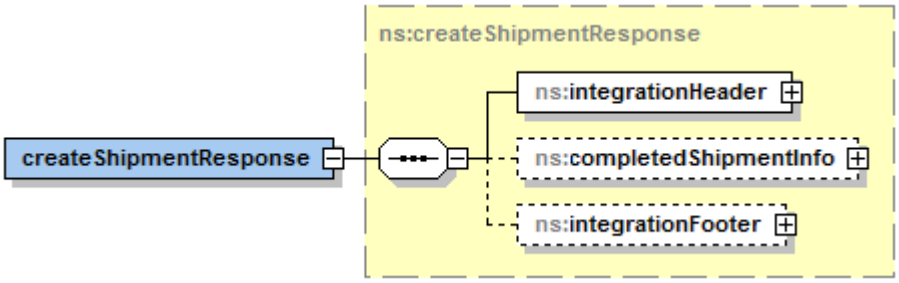


Table 19 – createShipmentResponse Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.4
completedShipmentInfo	N/A	0-1	element	Container for the completed shipment information. See section below for more information.
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the createShipment operation. Please see section 8.5 for more information.

completedShipmentInfo element

Figure 16 – completedShipmentInfo Structure

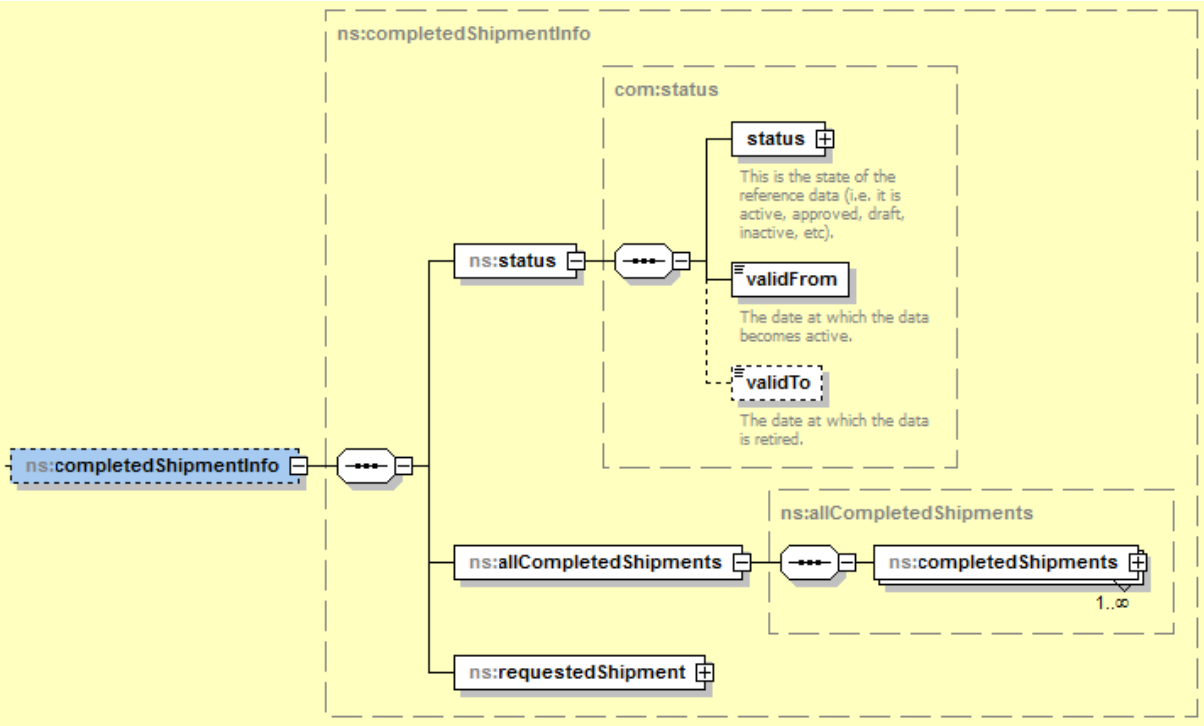


Table 20 – completedShipmentInfo Element

Element	Max Length	Occurs	Data Type	Details
status	N/A	1-1	element	Container for shipment status information
status/status	N/A	1-1	element	Container for shipment status details
../status/statusCode	N/A	1-1	element	Container for shipment status code
../../statusCode/code	9	1-1	identifier	This will be set to 'Allocated'. We can provide the full set of values the shipment status code can hold.
../status/validFrom	N/A	1-1	dateTime	Datetime value associated with when the shipment status code is valid from.
../status/validTo	N/A	0-1	dateTime	Datetime value associated with when the shipment status code is valid to. This is NOT USED.
allCompletedShipments	N/A	1-1	element	Container for completed shipments
allCompletedShipments/completedShipments	N/A	1- ∞	element	Container for completed shipments from processing the associated request.
../completedShipments/weight	N/A	0-1	element	The weight band the completed shipments are within
../../weight/value	5	1-1	float	Weight in grams of each of the items (no decimal places)
../../weight/unitofMeasure	N/A	1-1	element	Container for unit of measure
../../weight/unitofMeasure/unitofMeasureCode	N/A	1-1	element	Container for unit of measure code
../../weight/unitofMeasure/unitofMeasureCode/code	5	1-1	identifier	Must be 'g' for grams
../completedShipments/shipments	N/A	1- ∞	element	Container for completed shipment details

Element	Max Length	Occurs	Data Type	Details
../completedShipments/ shipments/shipmentNumber	13	0-99	identifier	For barcoded products, this field will contain the barcode number. For non-barcoded products, this field will contain the Shipping API internal reference number. For requests where there are multiple items, there will be a corresponding shipmentNumber for each item.
requestedShipment	n/a	1-1	element	The requestedShipment information as provided in the associated createShipmentRequest message. See section 8.7.2 for more information.

8.7.4 Example Data

Full XML examples of SOAP requests and responses are provided in Section 0 at the end of this document. This section provides a simplified example without the SOAP wrappers and associated security elements.

Example createShipment Request

```
<v1:createShipmentRequest>
  <v1:integrationHeader>
    <v11:identification>
      <v11:applicationId>111111113</v11:applicationId>
      <v11:transactionId>420642961</v11:transactionId>
    </v11:identification>
  </v1:integrationHeader>
  <v1:requestedShipment>
    <v1:shipmentType>
      <code>Delivery</code>
    </v1:shipmentType>
    <v1:serviceOccurence>4</v1:serviceOccurence>
    <v1:serviceType>
      <code>T</code>
    </v1:serviceType>
    <v1:serviceOffering>
      <serviceOfferingCode>
        <code>TPN</code>
      </serviceOfferingCode>
    </v1:serviceOffering>
    <v1:serviceFormat>
      <serviceFormatCode>
        <code>N</code>
      </serviceFormatCode>
    </v1:serviceFormat>
    <v1:bfpoFormat>
      <bFPOFormatCode/>
    </v1:bfpoFormat>
    <v1:serviceEnhancements>
      <v1:enhancementType>
        <serviceEnhancementCode>
          <code>SMS</code>
        </serviceEnhancementCode>
      </v1:enhancementType>
    </v1:serviceEnhancements>
    <v1:shippingDate>2014-08-16</v1:shippingDate>
    <v1:recipientContact>
      <v1:name>Mayor Janet Neetles</v1:name>
      <v1:complementaryName>Springfield Post Office</v1:complementaryName>
      <v1:telephoneNumber>
        <telephoneNumber>07123123123</telephoneNumber>
```

```
        </v1:telephoneNumber>
        <v1:electronicAddress>
          <electronicAddress>mayor.janet@springfield.com</electronicAddress>
        </v1:electronicAddress>
      </v1:recipientContact>
      <v1:recipientAddress>
        <addressLine1>Blackwell House</addressLine1>
        <addressLine2>123 Steep Street</addressLine2>
        <postTown>London</postTown>
        <postcode>SW2 5QR</postcode>
      </v1:recipientAddress>
      <v1:items>
        <v1:item>
          <v1:numberOfItems>1</v1:numberOfItems>
          <v1:weight>
            <unitOfMeasure>
              <unitOfMeasureCode>
                <code>g</code>
              </unitOfMeasureCode>
            </unitOfMeasure>
            <value>145</value>
          </v1:weight>
        </v1:item>
      </v1:items>
      <v1:departmentReference>3000447342</v1:departmentReference>
      <v1:customerReference>myCustRef</v1:customerReference>
      <v1:senderReference>mySenderRef</v1:senderReference>
    </v1:requestedShipment>
  </v1:createShipmentRequest>
```

Example createShipment Response

```
<NS1:createShipmentResponse>
  <NS1:integrationHeader>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>420642961</transactionId>
    </identification>
  </NS1:integrationHeader>
  <NS1:completedShipmentInfo>
    <NS1:status>
      <status>
        <statusCode>
          <code>Allocated</code>
        </statusCode>
      </status>
      <validFrom>2014-08-16T15:29:55</validFrom>
    </NS1:status>
    <NS1:allCompletedShipments>
      <NS1:completedShipments>
        <NS1:weight>
          <unitOfMeasure>
            <unitOfMeasureCode>
              <code>g</code>
            </unitOfMeasureCode>
          </unitOfMeasure>
          <value>145</value>
        </NS1:weight>
        <NS1:shipments>
          <NS1:shipmentNumber>JX002380709GB</NS1:shipmentNumber>
        </NS1:shipments>
      </NS1:completedShipments>
    </NS1:allCompletedShipments>
```

```
        <NS1:requestedShipment>
          <NS1:shipmentType>
            <code>Delivery</code>
          </NS1:shipmentType>
          <NS1:serviceOccurence>4</NS1:serviceOccurence>
          <NS1:serviceType>
            <code>T</code>
          </NS1:serviceType>
          <NS1:serviceOffering>
            <serviceOfferingCode>
              <code>TPN</code>
            </serviceOfferingCode>
          </NS1:serviceOffering>
          <NS1:serviceFormat>
            <serviceFormatCode>
              <code>N</code>
            </serviceFormatCode>
          </NS1:serviceFormat>
          <NS1:shippingDate>2014-08-16</NS1:shippingDate>
          <NS1:recipientContact>
            <NS1:name>Mayor Janet Neetles</NS1:name>
            <NS1:complementaryName>Springfield Post Office</NS1:complementaryName>
            <NS1:telephoneNumber>
              <telephoneNumber>07123123123</telephoneNumber>
            </NS1:telephoneNumber>
            <NS1:electronicAddress>
              <electronicAddress>mayor.janet@springfield.com</electronicAddress>
            </NS1:electronicAddress>
          </NS1:recipientContact>
          <NS1:recipientAddress>
            <addressLine1>Blackwell House</addressLine1>
            <addressLine2>123 Steep Street</addressLine2>
            <postTown>London</postTown>
            <postcode>SW2 5QR</postcode>
            <country>
              <countryCode>
                <code>GB</code>
              </countryCode>
            </country>
          </NS1:recipientAddress>
          <NS1:items>
            <NS1:item>
              <NS1:numberOfItems>1</NS1:numberOfItems>
              <NS1:weight>
                <unitOfMeasure>
                  <unitOfMeasureCode>
                    <code>g</code>
                  </unitOfMeasureCode>
                </unitOfMeasure>
                <value>145</value>
              </NS1:weight>
            </NS1:item>
          </NS1:items>
          <NS1:customerReference>myCustRef</NS1:customerReference>
          <NS1:senderReference>mySenderRef</NS1:senderReference>
        </NS1:requestedShipment>
      </NS1:completedShipmentInfo>
    </NS1:createShipmentResponse>
```

8.8 updateShipment operation

8.8.1 Description

The updateShipment operation allows customers to update the details of a previously created (but not manifested) shipment, provided that doing so does not violate the validation rules applied, or change the barcode number (e.g. it is not possible to update a Special Delivery item to become a Tracked Next Day item).

Only one shipment (identified by a single shipment number) can be updated per request, although multiple fields can be updated each time. If any field fails validation, then an error code will be returned and no fields will be updated. There is no limit to the number of times a shipment can be updated.

The status of the shipment is not affected by the update (e.g. a shipment with status 'Allocated' before an update will be 'Allocated' afterwards; a shipment with status 'Printed' will be 'Printed' afterwards)

8.8.2 updateShipment Request Message

To invoke the updateShipment operation, the customer shipping system constructs a SOAP request message (see section 7) with the updateShipmentRequest element contained within the SOAP Body. The updateShipmentRequest element is described in the figure and table below:

updateShipmentRequest element

Figure 17 – updateShipmentRequest Structure

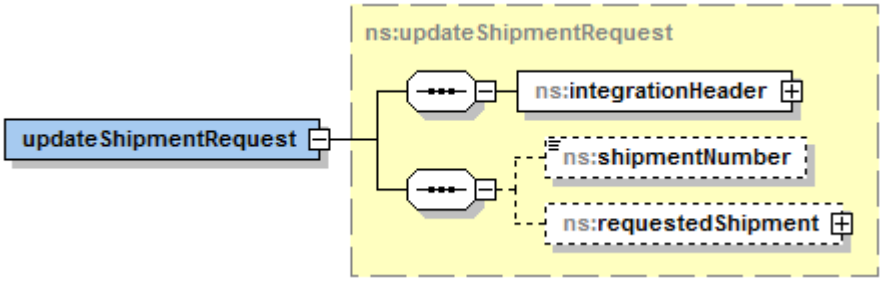


Table 21 – updateShipmentRequest Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.4
shipmentNumber	13	0-1	identifier	**Mandatory**. The number of the shipment to update.
requestedShipment	N/A	0-1	element	The updated details for the identified shipment. This is described for the createShipment operation in Section 8.7

8.8.3 updateShipmentResponse Message

The response for a requested updateShipment operation is constructed as a SOAP message with the updateShipmentResponse element contained in SOAP Body. The updateShipmentResponse element is described in the figure and table below:

updateShipmentResponse element

Figure 18 – updateShipmentResponse Structure

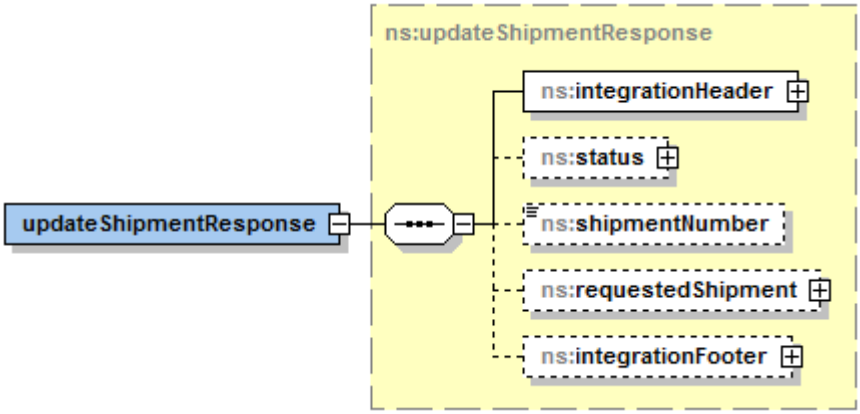


Table 22 – updateShipmentResponse Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	See section 8.4
status	N/A	1-1	element	Container for shipment status information
status/status	N/A	1-1	element	Container for shipment status details
../status/statusCode	N/A	1-1	element	Container for shipment status code
../../statusCode/code	9	1-1	identifier	This will be set to current status of the updated shipment: 'Allocated' or 'Printed'. Please see section 12.2 for the full set of values shipment status can have.
../status/validFrom	N/A	1-1	dateTime	Datetime value associated with when the shipment status code is valid from.
../status/validTo	N/A	0-1	dateTime	NOT USED. Datetime value associated with when the shipment status code is valid to.
shipmentNumber	13	0-1	identifier	The number of the shipment updated
requestedShipment	N/A	0-1	element	As described for the createShipment operation in section 8.7. Note that the barcode allocated to a shipment cannot be changed the updateShipment operation and therefore the serviceType and serviceEnhancements elements cannot be altered by the updateShipment operation. If changes to these elements are included in the requestedShipment element, an error will be returned.
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the updateShipment operation. Please see section 8.5 for more information.

8.8.4 Example Data

Full XML examples of SOAP requests and responses are provided in Section 0 at the end of this document. This section provides a simplified example without the SOAP wrappers and associated security elements.

Example updateShipment Request

```
<v1:updateShipmentRequest>
  <v1:integrationHeader>
    <v11:identification>
      <v11:applicationId>111111113</v11:applicationId>
      <v11:transactionId>420642961</v11:transactionId>
    </v11:identification>
  </v1:integrationHeader>
  <v1:shipmentNumber>JX002380709GB </v1:shipmentNumber>
  <v1:requestedShipment>
    <v1:shipmentType>
      <code>Delivery</code>
    </v1:shipmentType>
    <v1:serviceType>
      <code>1</code>
    </v1:serviceType>
    <v1:serviceOffering>
      <serviceOfferingCode>
        <code>CRL</code>
      </serviceOfferingCode>
    </v1:serviceOffering>
    <v1:serviceFormat>
      <serviceFormatCode>
        <code>L</code>
      </serviceFormatCode>
    </v1:serviceFormat>
    <v1:shippingDate>2014-01-07</v1:shippingDate>
        <v1:recipientContact>
      <v1:name>Johnny North</v1:name>
    </v1:recipientContact>
    <v1:recipientAddress>
      <addressLine1>132 Barnaby Way</addressLine1>
      <postTown>Chigwell</postTown>
      <postcode>IG99 6AA</postcode>
    </v1:recipientAddress>
    <v1:items>
      <v1:item>
        <v1:numberOfItems>1</v1:numberOfItems>
        <v1:weight>
          <unitOfMeasure>
            <unitOfMeasureCode>
              <code>g</code>
            </unitOfMeasureCode>
          </unitOfMeasure>
          <value>2</value>
        </v1:weight>
      </v1:item>
    </v1:items>
  </v1:requestedShipment>
</v1:updateShipmentRequest>
```

Example updateShipment Response

```
<NS1:updateShipmentResponse xmlns:NS1="http://www.royalmailgroup.com/api/ship/V1">
  <NS1:integrationHeader>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>420642961</transactionId>
    </identification>
  </NS1:integrationHeader>
  <NS1:status>
    <status>
      <statusCode>
        <code>Allocated</code>
      </statusCode>
    </status>
    <validFrom>2013-12-18T21:15:53</validFrom>
  </NS1:status>
  <NS1:shipmentNumber>JX002380709GB </NS1:shipmentNumber>
  <NS1:requestedShipment>
    <NS1:shippingDate>2013-12-18</NS1:shippingDate>
    <NS1:recipientContact>
      <NS1:name>John West</NS1:name>
    </NS1:recipientContact>
    <NS1:recipientAddress>
      <addressLine1>132 Barnaby Way</addressLine1>
      <postTown>Chigwell</postTown>
      <postcode>IG7 6NZ</postcode>
      <country>
        <countryCode>
          <code>GB</code>
        </countryCode>
      </country>
    </NS1:recipientAddress>
    <NS1:items>
      <NS1:item>
        <NS1:weight>
          <unitOfMeasure>
            <unitOfMeasureCode>
              <code>g</code>
            </unitOfMeasureCode>
          </unitOfMeasure>
          <value>2.000</value>
        </NS1:weight>
      </NS1:item>
    </NS1:items>
  </NS1:requestedShipment>
</NS1:updateShipmentResponse>
```

8.9 cancelShipment operation

8.9.1 Description

The cancelShipment operation allows customers to update the details of a previously created (but not manifested) shipment. Once a shipment has been cancelled its status will change from 'Allocated' or 'Printed' to 'Cancelled'.

Shipments created by the GUI or API can be cancelled by the API call, and cancelled shipments are visible in the system. Any Tracked Returns shipments must be cancelled before midnight as this is when they will be automatically processed and archived by the system.

Up to 1,000 shipments can be cancelled in a single request.

Any shipments that can't be cancelled will be communicated as an error message in the integrationFooter as described in section 8.5.

8.9.2 cancelShipmentRequest Message

To invoke the cancelShipment operation, the customer shipping system constructs a SOAP message (see section 7) with the cancelShipmentRequest element contained within the SOAP Body. The cancelShipmentRequest element is described in the figure and table below:

cancelShipmentRequest element

Figure 19 – cancelShipmentRequest Structure

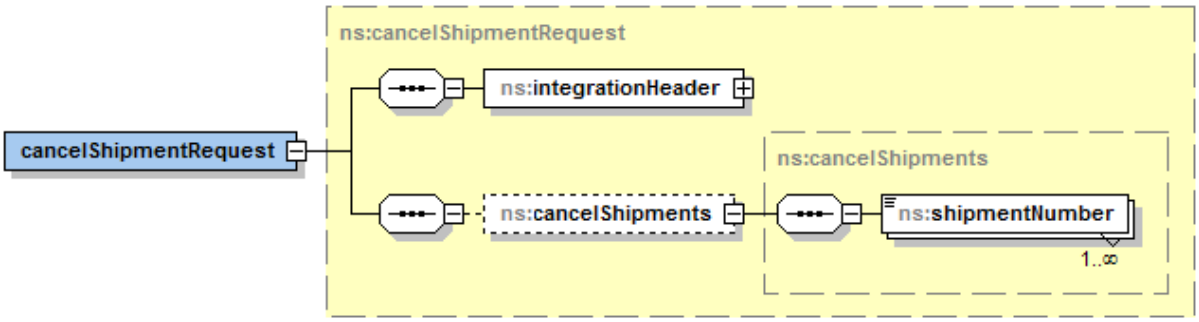


Table 23 – cancelShipmentRequest Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.4
cancelShipments	13	0-1	element	Container for the list of shipments to cancel
cancelShipments/ shipmentNumber	13	1- ∞	identifier	The shipment number to be cancelled

8.9.3 cancelShipmentResponse Message

The response for a requested updateShipment operation is constructed as a SOAP message described (see section 7) with the cancelShipmentResponse element contained within the SOAP Body. The cancelShipmentResponse element is described in the figure and table below:

cancelShipmentResponse element

Figure 20 – cancelShipmentResponse Structure

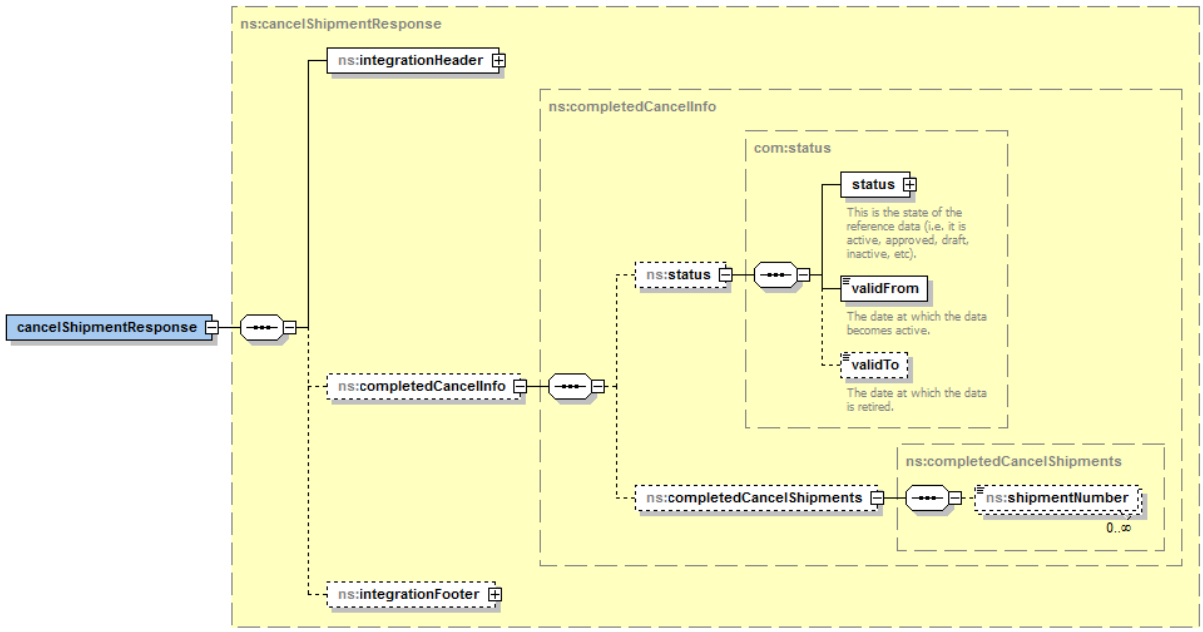


Table 24 – cancelShipmentResponse Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.4
completedCancelInfo	N/A	0-1	element	Container for the set of successfully cancelled shipments
completedCancelInfo/status	N/A	1-1	element	Container for shipment status information
../status/status	N/A	1-1	element	Container for shipment status details
../status/status/statusCode	N/A	1-1	element	Container for shipment status code
../status/status/statusCode/code	9	1-1	identifier	This will be set to 'Cancelled'. Please see section 12.2 for the full set of values shipment status can have.
../status/status/validFrom	N/A	1-1	dateTime	Datetime value associated with when the shipment status code is valid from.
../status/status/validTo	N/A	0-1	dateTime	NOT USED. Datetime value associated with when the shipment status code is valid to.
completedCancelInfo/ completedCancelShipments	N/A	0-1	element	Container for cancelled shipment numbers
../completedCancelShipments/ shipmentNumber	13	0- ∞	identifier	The number of each shipment that has been successfully cancelled.

Element	Max Length	Occurs	Data Type	Description
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the cancelShipment operation. Please see section 8.5 for more information.

8.9.4 Example Data

Full XML examples of SOAP requests and responses are provided in Section 0 at the end of this document. This section provides a simplified example without the SOAP wrappers and associated security elements.

Example cancelShipmentRequest

```
<v1:cancelShipmentRequest>
  <v1:integrationHeader>
    <v11:dateTime>2013-12-13T10:40:29</v11:dateTime>
    <v11:version>1.0</v11:version>
    <v11:identification>
      <v11:applicationId>111111113</v11:applicationId>
      <v11:transactionId>420642961</v11:transactionId>
    </v11:identification>
  </v1:integrationHeader>
  <v1:cancelShipments>
    <v1:shipmentNumber>JX002380709GB</v1:shipmentNumber>
  </v1:cancelShipments>
</v1:cancelShipmentRequest>
```

Example cancelShipmentResponse

```
<NS1:cancelShipmentResponse xmlns:NS1="http://www.royalmailgroup.com/api/ship/V1">
  <NS1:integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2013-12-
13T10:40:29</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">1</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>420642961</transactionId>
    </identification>
  </NS1:integrationHeader>
  <NS1:completedCancelInfo>
    <NS1:status>
      <status>
        <statusCode>
          <code>Cancelled</code>
        </statusCode>
      </status>
      <validFrom>2013-12-13T13:48:27</validFrom>
    </NS1:status>
    <NS1:completedCancelShipments>
      <NS1:shipmentNumber>JX002380709GB</NS1:shipmentNumber>
    </NS1:completedCancelShipments>
  </NS1:completedCancelInfo>
  <NS1:integrationFooter/>
</NS1:cancelShipmentResponse>
```

8.10 printLabel operation

8.10.1 Description

The printLabel operation allows customers to request a label in Base64 encoded PDF format for a specific shipment. Once the printLabel operation has been called on a shipment with status 'Allocated', the status for that shipment is changed to 'Printed'.

Shipments created by either the GUI or API can be printed by the API call. There is no limit on the number of times the PrintLabel request can be used on a shipment, though high numbers of reprints will be flagged to Royal Mail.

8.10.2 printLabelRequest Message

To invoke the printLabel operation, the customer shipping system constructs a SOAP message (see section 7) with the printLabelRequest element contained within the SOAP Body. The printLabelRequest element is described in the figure and table below:

printLabelRequest element

Figure 21 – printLabelRequest Structure

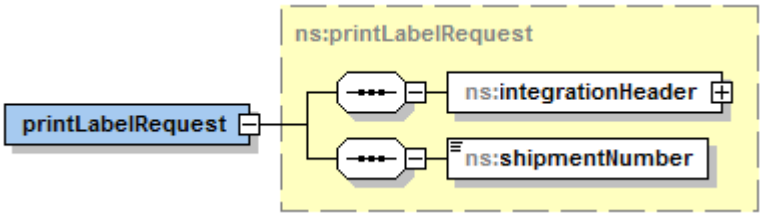


Table 25 – printLabelRequest Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.4
shipmentNumber	13	1-1	identifier	The shipment number of the shipment to be printed

8.10.3 printLabelResponse Message

The response for a requested printLabel operation is constructed as a SOAP message described (see section 7) with the printLabelResponse element contained within the SOAP Body. The printLabelResponse element is described in the figure and table below.

printLabelResponse element

Figure 22 – printLabelResponse Structure

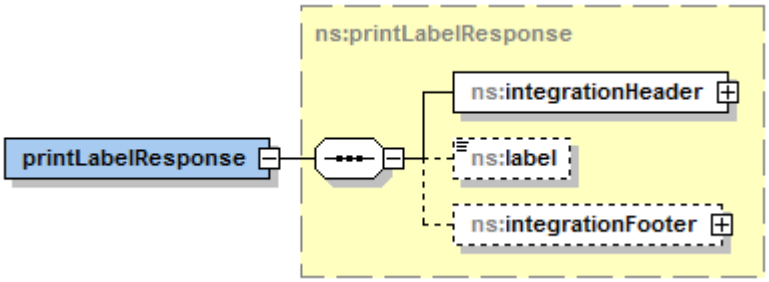


Table 26 – printLabelResponse Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.4
label		0-1	document	Label in PDF format and Base64 encoded
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the printLabel operation. Please see section 8.5 for more information.

8.10.4 Example Data

Full XML examples of SOAP requests and responses are provided in Section 0 at the end of this document. This section provides a simplified example (which has been truncated) without the SOAP wrappers and associated security elements.

Example printLabelRequest

```
<v1:printLabelRequest>
  <v1:integrationHeader>
    <v11:dateTime>2014-02-27T12:49:45</v11:dateTime>
    <v11:version>1.0</v11:version>
    <v11:identification>
      <v11:applicationId>111111113</v11:applicationId>
      <v11:transactionId>420642961</v11:transactionId>
    </v11:identification>
  </v1:integrationHeader>
  <v1:shipmentNumber> JX002380709GB</v1:shipmentNumber>
</v1:printLabelRequest>
```

Example printLabelResponse

```
<printLabelResponse xmlns="http://www.royalmailgroup.com/api/ship/V1">
  <integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2013-12-
04T16:55:29</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">1</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>420642961</transactionId>
    </identification>
```

```
        </integrationHeader>
        <label>JVBERi0xLjYKJeTjz9IKMSAwIG9iagpbL1BERi9JbWFnZUIvSW1hZ2VDL0ltYWdlSS9UZXh0XQpI
bmRvYmoKNCAwIG9iago8PC9MZW5ndGggNSAwIFIKL0ZpbHRlci9GbGF0ZURlY29kZQo+PgpzdHJI
YW0KeJwzUDCAwxRDLgATpwKACmVuZHN0cmVhbQplbmRvYmoKNSAwIG9iagoxNQplbmRvYmoKNiAw
IG9iago8PC9MZW5ndGggNyAwIFIKL0ZpbHRlci9GbGF0ZURlY29kZQo+PgpzdHJlYW0KeJwzNTBQ
MCBuIAowMDAwMDQ5NjM3IDAwMDAwIG4gCjAwMDAwNTAyNDUgMDAwMDAgbiAKMDAwMDA1NDA4NyAw
MDAwMCBuIAowMDAwMDUzNzAyIDAwMDAwIG4gCjAwMDAwNTU0NDIgMDAwMDAgbiAKMDAwMDA1NTg2
NyAwMDAwMCBuIAp0cmFpbGVyCjw8L1NpemUgNzMKL1Jvb3QgNzIgMCBSCi9JbmZvIDY3IDAgUgov
SURbPDg1MkYxMEIzQzQ0QTk5RTE3MEE3NUM5NjhDQTQyRUI1Pjw4NTJGMTBCM0M0NEE5OUUxNzBB
NzVDOTY4Q0E0MkVCNT5dCj4+CnN0YXJ0eHJlZgo1NTk0MAolJUVPRgo=
        </label>
    </printLabelResponse>
```


8.11 createManifest operation

8.11.1 Description

The createManifest operation allows customers to submit to Royal Mail details of all of the items that will be despatched that day. Once the createManifest operation has been called, all shipments that have status 'Printed' will be set to status 'Manifested' and it will no longer be possible to update or cancel them.

Manifests can be created by Service Reference or by Service Code, or if neither is specified then all shipments that have status 'Printed' will be included (N.B. Tracked Returns are not included in any part of the manifesting process).

8.11.2 createManifestRequest Message

To invoke the createManifest operation, the customer shipping system constructs a SOAP message (see section 7) with the createManifestRequest element contained within the SOAP Body. The createManifestRequest element is described in the figure and table below:

createManifestRequest element

Figure 23 – createManifestRequest Structure

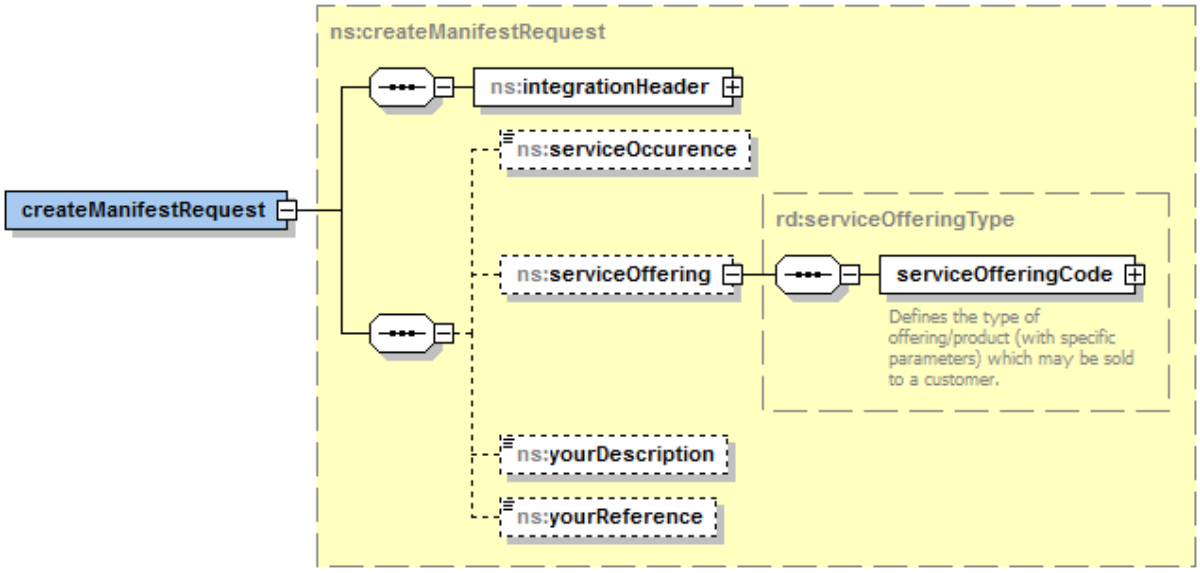


Table 27 – createManifestRequest Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.4
serviceOccurence	2	0-1	ordinal	All or one Service Occurrence. If not included then ALL Service Occurrences are included in the manifest.
serviceOffering	N/A	0-1	element	Container for service offering details
serviceOffering/ serviceOfferingCode	N/A	1-1	element	Container for service offering code
../serviceOfferingCode/code	3	0-1	identifier	All or one Service Offerings. If not included then ALL Service Offerings are included in the manifest. Please note that this field is case sensitive.
yourDescription	40	0-1	comment	This is a description field that corresponds to the Your Description field in RM's Online Business Account (OBA). This is for customer reference and will not appear on any paperwork.
yourReference	40	0-1	identifier	Included on the Customer Collection Receipt.

8.11.3 createManifestResponse Message

The response for a requested createManifest operation is constructed as a SOAP message (see section 7) with the createManifestResponse element contained within SOAP Body. The createManifestResponse element is described in the figure and table below:

createManifestResponse element

Figure 24 – createManifestResponse Structure

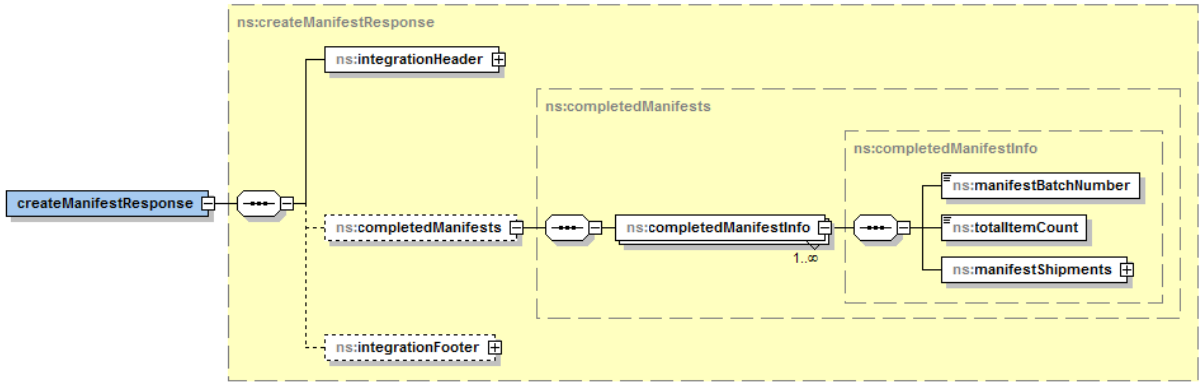


Table 28 – createManifestResponse Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in Section 8.4
completedManifests	N/A	0-1	element	Container for completed manifest details
completedManifests/ completedManifestInfo	N/A	1- ∞	element	Container for information on the manifest that was created
../completedManifestInfo/ manifestBatchNumber	4	1-1	identifier	The batch number of the manifest. Used in subsequent call to the printManifest operation
../completedManifestInfo/ totalItemCount	2	1-1	cardinal	Total number of shipments on this manifest

Element	Max Length	Occurs	Data Type	Description
../completedManifestInfo/ manifestShipments	N/A	1-1	element	Container for the list of shipments included in the manifest
../../manifestShipments/ manifestShipment	N/A	1- ∞	element	Container for manifest shipment information
../../../manifestShipment/serviceOffering	N/A	1-1	element	Container for service offering details
../../../manifestShipment/serviceOffering/ serviceOfferingCode	N/A	1-1	element	Container for service offering code
../../../manifestShipment/serviceOffering/ serviceOfferingCode/code	N/A	1-1	identifier	The service offering code of a shipment on the manifest
../../manifestShipments/ manifestShipment/shipmentNumber	13	1-1	identifier	The shipment number of a shipment on the manifest
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the createManifest operation. Please see section 8.5 for more information.

8.11.4 Example Data

Full XML examples of SOAP requests and responses are provided in Section 0 at the end of this document. This section provides a simplified example (which has been truncated) without the SOAP wrappers and associated security elements.

Example createManifestRequest

```
<v1:createManifestRequest>
  <v1:integrationHeader>
    <dateTime>2014-01-01T21:32:52</dateTime>
    <v11:version>1.0</v11:version>
    <v11:identification>
      <v11:applicationId>111111113</v11:applicationId>
      <v11:transactionId>420642961</v11:transactionId>
    </v11:identification>
  </v1:integrationHeader>
  <v1:serviceOffering>
    <serviceOfferingCode>
      <code>STL</code>
    </serviceOfferingCode>
  </v1:serviceOffering>
  <v1:yourDescription>Shipments140213</v1:yourDescription>
  <v1:yourReference>Shipments140213</v1:yourReference>
</v1:createManifestRequest>
```

Example createManifestResponse

```
<NS1:createManifestResponse xmlns:NS1="http://www.royalmailgroup.com/api/ship/V1">
  <NS1:integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2014-01-
01T21:32:52</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">1</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>420642961</transactionId>
    </identification>
    <testFlag xmlns="http://www.royalmailgroup.com/integration/core/V1">true</testFlag>
    <debugFlag xmlns="http://www.royalmailgroup.com/integration/core/V1">true</debugFlag>
```

```
    </NS1:integrationHeader>
    <NS1:completedManifests>
      <NS1:completedManifestInfo>
        <NS1:manifestBatchNumber>190</NS1:manifestBatchNumber>
        <NS1:totalItemCount>3</NS1:totalItemCount>
        <NS1:manifestShipments>
          <NS1:manifestShipment>
            <NS1:serviceOffering>
              <serviceOfferingCode>
                <code>TPS</code>
              </serviceOfferingCode>
            </NS1:serviceOffering>
            <NS1:shipmentNumber>JC924043946GB</NS1:shipmentNumber>
          </NS1:manifestShipment>
                <NS1:manifestShipment>
            <NS1:serviceOffering>
              <serviceOfferingCode>
                <code>TPS</code>
              </serviceOfferingCode>
            </NS1:serviceOffering>
            <NS1:shipmentNumber>JC924043947GB</NS1:shipmentNumber>
          </NS1:manifestShipment>
                <NS1:manifestShipment>
            <NS1:serviceOffering>
              <serviceOfferingCode>
                <code>TPS</code>
              </serviceOfferingCode>
            </NS1:serviceOffering>
            <NS1:shipmentNumber>JC924043948GB</NS1:shipmentNumber>
          </NS1:manifestShipment>
        </NS1:manifestShipments>
      </NS1:completedManifestInfo>
    </NS1:completedManifests>
    <NS1:integrationFooter/>
  </NS1:createManifestResponse>
```

8.12 printManifest operation

8.12.1 Description

The printManifest operation allows customers to request a manifest (Customer Collection Receipt) in Base64 encoded PDF format. Once the printManifest operation has been called, all shipments that have status 'Manifested' will be set to status 'ManifestedPrinted'. Manifests can be specified by either the manifestBatchNumber or the salesOrderNumber (the salesOrderNumbers are available via the GUI the day after the manifest was created).

There is no limit on the number of times the PrintManifest request can be used on a manifest, however reprints will not include the barcodes and excessive requests for reprints will be flagged to Royal Mail.

8.12.2 printManifestRequest Message

To invoke the printManifest operation, the customer shipping system constructs a SOAP message (see section 7) with the printManifestRequest element contained within the SOAP Body. The printManifestRequest element is described in the figure and table below:

printManifestRequest element

Figure 25 – printManifestRequest Structure

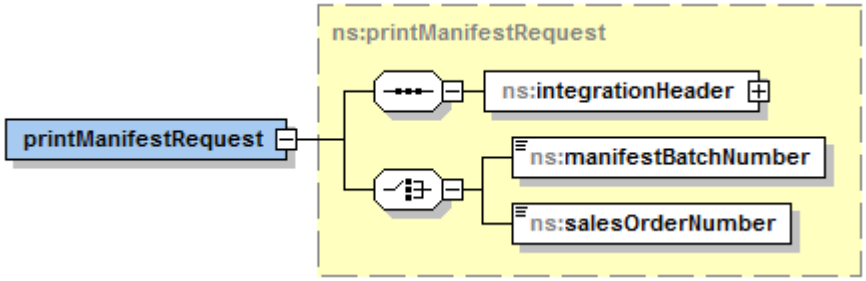


Table 29 – printManifestRequest Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.4
manifestBatchNumber	20	1-1	identifier	This is the batch number to print and is returned by a prior call to createManifest operation.
salesOrderNumber	20	1-1	identifier	The Sales Order Number, which is available via the GUI the day after the manifest was created.

8.12.3 printManifestResponse Message

The response for a requested printManifest operation is constructed as a SOAP message (see section 7) with the printManifestResponse element contained within the SOAP Body. The printManifestResponse element is described in the figure and table below:

printManifestResponse element

Figure 26 – printManifestResponse Structure

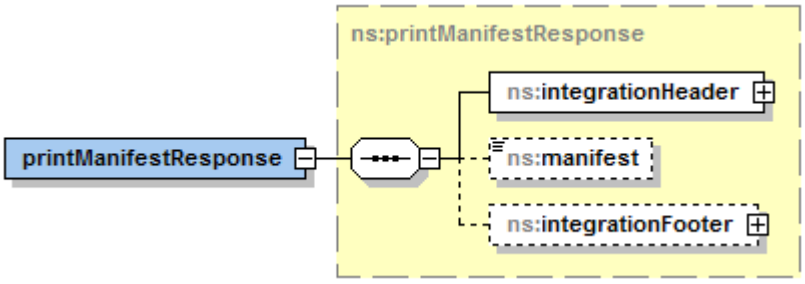


Table 30 – printManifestResponse Element

Element	Max Length	Occurs	Data Type	Description
integrationHeader	N/A	1-1	element	As described in section 8.4
manifest		0-1	document	Customer Collection Receipt in PDF format -Base64 encoded for transfer
integrationFooter	N/A	0-1	element	Container for any error or warning messages associated with the printManifest operation. Please see section 8.5 for more information.

8.12.4 Example Data

Full XML examples of SOAP requests and responses are provided in Section 0 at the end of this document. This section provides a simplified example (which has been truncated) without the SOAP wrappers and associated security elements.

Example printManifestRequest

```
<v1:printManifestRequest>
  <v1:integrationHeader>
    <v11:dateTime>2013-12-13T15:29:42</v11:dateTime>
    <v11:version>1.0</v11:version>
    <v11:identification>
      <v11:applicationId>111111113</v11:applicationId>
      <v11:transactionId>420642961</v11:transactionId>
    </v11:identification>
  </v1:integrationHeader>
  <v1:manifestBatchNumber>6</v1:manifestBatchNumber>
</v1:printManifestRequest>
```

Example printManifestResponse

```
<printManifestResponse xmlns="http://www.royalmailgroup.com/api/ship/V1">
  <integrationHeader>
    <dateTime xmlns="http://www.royalmailgroup.com/integration/core/V1">2013-12-
13T15:29:42</dateTime>
    <version xmlns="http://www.royalmailgroup.com/integration/core/V1">1</version>
    <identification xmlns="http://www.royalmailgroup.com/integration/core/V1">
      <applicationId>111111113</applicationId>
      <transactionId>420642961</transactionId>
    </identification>
  </integrationHeader>
  <manifest>JVBERi0xLjMKJeLjz9MKMSAwIG9iajw8L1Byb2R1Y2VyKGh0bWxkb2MgMS44LjI3IENvcHlyaWdo
dCAxOTk3LTIwMDYgRWFzeSBTb2Z0d2FyZSBQcm9kdWN0cywgQWxsIFJpZ2h0cyBSZXNlcnZlZC4p
L0NyZWF0aW9uRGF0ZShEOjIwMTMxMjEzMTUyMDU3LTAxMDApPj5lbmRvYmoKMiAwIG9iajw8L1R5
cGUvRW5jb2RpbmcvRGlmZmVyZW5jZXNbIDMyL3NwYWNlL2V4Y2xhbS9xdW90ZWRibC9udW1iZXJz
aWduL2RvbGxhci9wZXJjZW50L2FtcGVyc2FuZC9xdW90ZXNpbmdsZS9wYXJlbmxlZnQvcGFyZW5y
aWdodC9hc3Rlcmlzay9wbHVzL2NvbW1hL2h5cGhlbi9wZXJpb2Qvc2xhc2gvemVyby9vbmUvdHdv
L3RocmVlL2ZvdXIvZml2ZS9zaXgvc2V2ZW4vZWlnaHQvbmluZS9jb2
  </manifest>
</printManifestResponse>
```

9 Error Handling

9.1 Overview

The Shipping API service can communicate issues in three ways:

1. Technical Errors
   Highlights fundamental problems with either the system or the request. All technical errors (e.g. schema validation failure, service unavailable etc.) are returned as SOAP Faults to the service requester.
2. Business Errors
   While the request was correctly formatted; it contained invalid data that cannot be automatically corrected. All business errors are returned in the integrationFooter of the SOAP response message.
3. Warnings
   While the request was correctly formatted and the data was mostly accurate, there were minor corrections or assumptions that the system had to make to allow the request to be processed. All warnings are returned in the integrationFooter of the SOAP response message.

Errors and warnings are the same in both the onboarding and live environments. Please refer to the Shipping API V1 page on the Royal Mail Developer Portal for the full list of all error and warning codes which can be returned.

9.2 Technical Errors

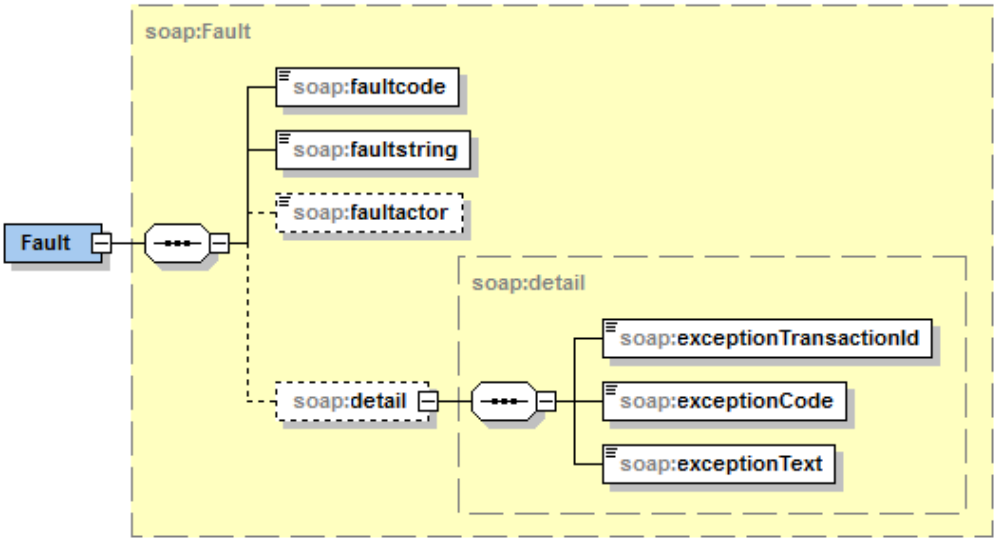
Technical Errors highlight that there is either a problem with the Royal Mail system or there is a fundamental problem with the messages being sent to Royal Mail by the customer. Examples of technical errors caused by the customer would include the submission of a request message that failed schema validation.

Technical errors indicate that the request was not successful, and has not been processed by Royal Mail (e.g. createShipment requests that receive a Technical Error will not result in a shipment being created on the system).

Technical errors may also be an indication of serious problems with the interaction between the customer and Royal Mail, and customer systems should be built in such a way that these errors are gracefully handled, captured and reported to the relevant technical resources. Failure to do so may result in a disruption to service.

All technical exceptions are returned to the customer using the SOAP Fault message construct. Please see figure below for a graphical representation of the SOAP Fault construct which has been extended by the Royal Mail Group to specify a number of sub-elements under the <detail> element.

Figure 27 – SOAP Fault Structure



The SOAP Fault element has the following sub-elements:

Table 31 – SOAP Fault Elements

Element	Max Length	Occurs	Data Type	Description
faultcode	N/A	1-1	QName	A code for identifying the fault
faultstring	N/A	1-1	string	A human readable explanation of the fault
faultactor	N/A	0-1	anyURI	Information about who caused the fault to happen.
detail	N/A	0-1	element	Container for application specific error information
detail/exceptionTransactionId	50	1-1	identifier	Used to identify the transaction Id associated with the request message which generated this technical exception.
detail/exceptionCode	5	1-1	identifier	Error code associated with the technical exception.
detail/exceptionText	256	1-1	description	A meaningful human-readable error description of the error condition.

We can provide the full set of technical errors that may be returned by the API and the values which will be populated in the above fields for each error condition.

9.3 Business Errors

Business errors indicate that the data provided in the request, while correctly formatted, is not valid. Examples of issues that would lead to a business error being received include attempting to use a service that the customer does not have an agreement line for, specifying a shipment date too far in the future, or attempting to update a shipment that has already been manifested.

Business errors indicate that the request was not successful, and has not been processed by Royal Mail (e.g. updateShipment requests that receive a business error will not result in any changes to the specified shipment).

All business errors are returned in the integrationFooter of the SOAP response message. Please see the 'Shipping API Reference Data' document on the Shipping API V1 page on the Royal Mail Developer Portal for the set of business errors that may be returned by the API.

9.4 Warnings

Warning messages are received when the data supplied to Royal Mail in the request is not completely accurate, but the system is able to make assumptions or corrections automatically. Examples of problems that would result in a warning message include: address lines with too many characters (that are then truncated), a telephone number being specified without an SMS option being selected (and hence being ignored), and a shipment date being specified in the past (and being automatically updated to today's date).

Requests that result in a warning message are processed by the system (e.g. a createShipment request where the customer reference is longer than 12 characters will be created on the system, but with the customer reference truncated to 12 characters).

All warnings are returned in the integrationFooter of the SOAP response message. Please see the 'Shipping API Reference Data' document on the Shipping API V1 page on the Royal Mail Developer Portal for the set of warnings that may be returned by the API.

10 Non Functional Characteristics

10.1 Availability

10.1.1 Service Hours

The Shipping API is available 24 hours per day x 365 days per year.

10.1.2 Maintenance Windows

Royal Mail Online Services Terms and Conditions define the maintenance window for this service.

10.1.3 Unavailability

In the unlikely event of the Shipping API being unavailable, customer systems should be able to handle this appropriately. Royal Mail will endeavour to proactively contact customers in the event of an outage to this API.

If you experience issues with the availability of this API please contact us at rmdmo@royalmail.com or 08456047267.

10.2 Performance

Performance will be slower during peak periods of activity (between 3pm and 6pm Monday to Friday). To avoid issues associated with slower response times during periods of heavy traffic, customers are strongly advised to spread their traffic out throughout the day (wherever possible).

10.3 Security

All API service calls will be made using mutually authenticated HTTPS bound SOAP web services. The Local Collect API exposes the services using SOAP version 1.1 with a document / literal document-style encoding.

All service requests via the API Management solution will be authorised in accordance with the Client ID and Secret passed in the HTTP headers. This will ensure that any external service requests are authorised and authenticated in line with RMG Security Policies and Standards.

11 Frequently Asked Questions

Please see the FAQ page on the Developer Portal for a general list of frequently asked questions with responses.

Please see the Shipping API API page on the Royal Mail Developer Portal for a list of frequently asked questions with responses.

12 Appendix A – Reference Data

Please see the 'Shipping API Reference Data' document on the Shipping API V2 page on the Royal Mail Developer Portal for the set of reference data which is used by this API.

12.1 Allowable Character Set

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	' '
010 0011	043	35	23	#
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1011	053	43	2B	+
010 1100	054	44	2C	,
010 1101	055	45	2D	-
010 1110	056	46	2E	.
010 1111	057	47	2F	/
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
011 0011	063	51	33	3
011 0100	064	52	34	4
011 0101	065	53	35	5
011 0110	066	54	36	6
011 0111	067	55	37	7
011 1000	070	56	38	8
011 1001	071	57	39	9
011 1010	072	58	3A	:
011 1111	077	63	3F	?

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1101	135	93	5D	]
101 1111	137	95	5F	_

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q
111 0010	162	114	72	r
111 0011	163	115	73	s
111 0100	164	116	74	t
111 0101	165	117	75	u
111 0110	166	118	76	v
111 0111	167	119	77	w
111 1000	170	120	78	x
111 1001	171	121	79	y
111 1010	172	122	7A	z
111 1011	173	123	7B	{
111 1100	174	124	7C	|
111 1101	175	125	7D	}
111 1110	176	126	7E	~

12.2 Shipment Status Codes

The values for a shipment status are defined in the table below:

Table 32 – Shipment Status Codes

<status>	Description
Allocated	Shipment with a Service Type / Service / Service Format and shipment number but not printed
Printed	Shipment with Service Type / Service / Service Format and shipment number and the label(s) printed
Manifested	Customer Collection Receipt has been created Customer Collection Receipt has been printed
Cancelled	Shipment has been cancelled

13 Appendix B – XML Examples

Please see the Shipping API V1 API page on the Royal Mail Developer Portal to download the following XML examples for each web service operation.

13.1 createShipment:

i. Request (createShipmentRequest.xml)
ii. Response (createShipmentResponse.xml)

13.2 updateShipment

i. Request (updateShipmentRequest.xml)
ii. Response (updateShipmentResponse.xml)

13.3 cancelShipment

i. Request (cancelShipmentRequest.xml)
ii. Response (cancelShipmentResponse.xml)

13.4 printLabel

i. Request (printLabelRequest.xml)
ii. Response (printLabelResponse.xml)

13.5 createManifest

i. Request (createManifestRequest.xml)
ii. Response (createManifestResponse.xml)

13.6 printManifest

i. Request (printManifestRequest.xml)
ii. Response (printManifstResponse.xml)