Get Documents Web Service- 1.1.0

Confidential

Revision Notes

Revision	Date	Change Description
1.0.0	04/23/2009	Version 1.0 Created.
1.0.1	11/27/2009	URL for endpoints has been modified. Version allowed value has been updated.
1.1.0	04/20/2010	Addition of GetShipmentManifestDocument Method Update to the Introduction. Update to Implementation Considerations.

Table of Contents

Legal and Copyright Notices

This document and the information contain herein is confidential and proprietary to Purolator Courier Ltd., and its use is governed by the Purolator E-Ship™ Resource Centre Agreement or the Purolator E-Ship™ Web Services Soft Launch Agreement (as applicable). In the event of a conflict between this document and the Purolator E-Ship Resource Centre Agreement or the Purolator E-Ship Web Services Soft Launch Agreement (as applicable), the Purolator E-Ship Resource Centre Agreement or the Purolator E-Ship Web Services Soft Launch Agreement (as applicable) will govern to the extent of the inconsistency.

The information in this documentation may be changed at any time without notice. If you have gained access to this document without first agreeing to the Purolator E-Ship Resource Centre Agreement or the Purolator E-Ship Web Services Soft Launch Agreement (as applicable) you must cease using this document immediately and destroy any copies obtained.

1 Getting Started

1.1 Introduction

To help you integrate Purolator E-Ship Web Services into your website or application, Purolator provides technical documentation and sample code for each Web Service. The documents describe how to consume each Web Service and detail Purolator's service offerings. In addition to this document, additional information and support is available on the Developers Forum http://www.purolatorwebservices.com and also on the E-Ship Web Services Wiki http://www.purolatorwebservices.com/wiki2.

The following documents can be downloaded by registered Purolator E-Ship Resource Centre users in the Documentation and Sample Code section.

Introduction
Introduction to Purolator E-Ship Web Services

Service Availability Web Service
Validate the origin and destination addresses and receive all available Purolator services, products, options and associated rules.

Estimates Web Service
Receive estimates based on origin and destination addresses as well as package and service inputs. Estimates are available as quick estimates using list prices or detailed estimates for customers with negotiated pricing.

Shipping Web Service
Create domestic, U.S. and international shipments. Please note that shipping labels and documents are created using the Get Documents Web Service.

Pickup Web Service
Schedule/Modify/Delete/View history for pickups.

Returns Management Web Service
Create domestic Returns Management shipments. Please note that shipping labels and documents are created using the Get Documents Web Service.

Get Documents Web Service
Retrieve shipment labels and documentation for Purolator shipments and Returns Management shipments.

Tracking Web Service
Get detailed tracking information for all Purolator shipments.

Test Cases
Test cases and expected results for all Purolator E-Ship Web Services.

The following documents will be provided when you are ready to take your website or application into Purolator's production environment.

Production Readiness
Guidelines to help ensure your Purolator E-Ship Web Services integration is ready to promote into Purolator's production environment.

Security Options
Security options available for your Purolator E-Ship Web Services implementation.

1.2 Document Overview

This document is a technical specification that describes the Purolator Get Documents Web Service. It is geared towards developers who are building applications that require Purolator shipping documentation. This includes for example, shipping labels that need to be affixed to each shipment package; Commercial Invoices that must accompany U.S. and International Shipments; NAFTA and FDA declaration forms for certain kinds of U.S. destined shipments.

The Get Documents Web Service is not a stand alone web service but is used in conjunction with other web services. For example, once a shipment has been created using the Shipping Web Service or Returns Management Web Service then Get Documents would be used to obtain shipping labels. For a more thorough description on the shipping process please see the 'Purolator E-Ship Web Services- Introduction' document at http://www.purolator.com/eship/documentation.

This document uses the following structure to explain Purolator's E-Ship Get Documents Web Service:

Introduction (this chapter)
- Web Services overview
- Specification compliance and development platform compatibility
- Related resources

Web Services
- Explanations of the characteristics of calls, as well as data types used
- Security information
- Service details for Get Documents service
- Implementation considerations
- Error details and error codes and descriptions

Appendix
- Complete field listing
- Enumeration and expected values

1.3 Specification Compliance

The Web Services are designed to comply with the following specifications:

Specification Name	Website
Simple Object Access Protocol (SOAP) 1.1	http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
Web Service Description Language (WSDL) 1.1	http://www.w3.org/TR/2001/NOTE-wsdl-20010315

Please visit the websites listed above for detailed information regarding SOAP and WSDL technologies.

1.4 Development Platforms

The Web Services work with any current SOAP/XML development environments, including, but not limited to, Visual Studio .NET 2005, JAVA, and PHP.

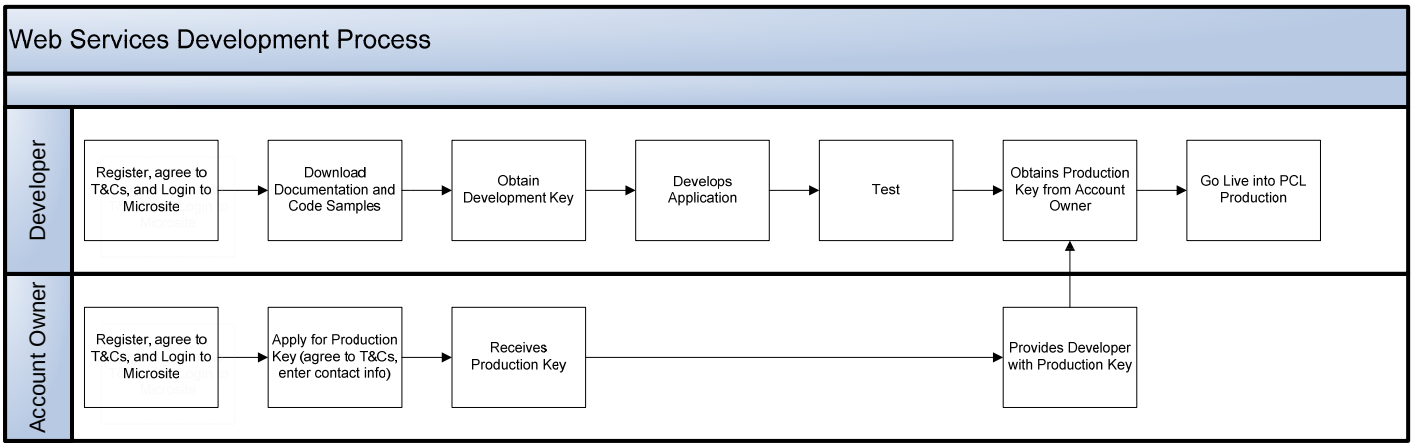
Note: Development platforms vary in their SOAP/XML implementations. Please refer to your platform's documentation for more information.

1.5 Related Resources

The Purolator E-Ship Developer Centre provides a full suite of documentation, sample code, sample SOAP messages, community-based support, and other resources to help you with your development projects. Be sure to visit http://eship.purolator.com to signup for a developer account.

1.6 Purolator E-Ship™ Web Services Development Process

To integrate Purolator E-Ship Web Services within your application, follow the steps outlined below. After registering for the Purolator E-Ship™ Resource Centre, both the Purolator account holder and software developer will have access to all resources on the Resource Centre website.



Web Services Developer Process

Step 1: **Register**

Sign up for the Purolator E-Ship Resource Centre to access all Web Services' resources.

Step 2: **Download Documentation and Code Samples**

View code samples and documentation for the Purolator E-Ship Web Service(s) you will be using.

Sample code such as; C#, Java, PHP, SOAP/XML as well as WSDL is available and can be found at the Purolator E-Ship Developer Resource Centre at: http://www.purolator.com/eship/samplecode.

Documentation, including testing documentation is available at http://www.purolator.com/eship/documentation.

Step 3: **Obtain Development Key**

Apply for a Development Key. This will allow you to test your code in the Purolator E-Ship Test Environment.

Apply for a development key (including a development password, and test account number) at http://www.purolator.com/eship/development.

Note: The key, account number and password will be used to access the development environment at https://devwebservices.purolator.com/EWS/V1/ShippingDocuments/ShippingDocumentsService.asmx.

Step 4: **Develop**

Develop your application using the code samples and documentation.

Step 5: **Test**

Test your application in the Purolator E-Ship Test Environment. Data in the Test Environment is sample data and should be used for testing purposes only. Recommended test cases with expected results are provided in the documentation.

Step 6: **Obtain Production Key from Purolator Account Holder**

When your development and testing are complete, replace the Development Key with the Production Key. You can obtain the Production Key from the Purolator account holder. Once activated, the Production Key will allow you to access account-specific shipping, tracking, estimates and returns information. Optional security features are available through consultation with Purolator.

Note: The key, account number and password will be used to access the production environment at https://webservices.purolator.com/EWS/V1/ShippingDocuments/ShippingDocumentsService.asmx.

Step 7: **Go Live**

Launch your application with Purolator E-Ship Web Services.

Purolator Account Holder Process

Step 1: **Register**

Sign up for the Purolator E-Ship™ Resource Centre to access all Web Services' resources.

Step 2: **Apply for a Production Key**

To use Purolator E-Ship™ Web Services, you must apply for a Production Key. The Production Key will allow you to access shipping, tracking, estimates and returns information specific to your Purolator account. A Purolator representative may contact you for security purposes before your Production Key is activated. Optional security features are available through consultation with Purolator.

Apply for a production key (including a production password) at http://www.purolator.com/eship/production.

Step 3: **Provide your developer with the Production Key**

Ask your developer to replace their Development Key with your Production Key. You will then be ready to use Purolator E-Ship Web Services.

Step 4: **Go Live**

Launch your application with Purolator E-Ship Web Services.

2 Web Services

Web Services is a collection of programming technologies, including XML, Web Services Description Language (WSDL) and SOAP, which allow you to build programming solutions for specific messaging and application integration.

Web Services are, by definition, platform independent. Purolator Web Services allow developers to build custom applications that are independent of changes to the Purolator interface. This is achieved through backwards compatibility support for all versions of the web services.

2.1 Characteristics of Calls

All Web Service calls are:

- **Service Requests and Responses**—your client application prepares and submits a service request to Purolator E-Ship Web Services, which is processed and a response is returned, and the client application handles the response.

- **Synchronous**—once the API call is invoked, your client application waits until it receives a response from the service. Asynchronous calls are not supported.

2.2 Data Types

Value	Description
String	Character strings. Fields that are of data type string contain text and some have length restrictions. Please see the field reference list for format and restrictions.
Decimal	Decimal represents a subset of the real numbers, which can be represented by decimal numerals. The ·value space· of decimal is the set of numbers that can be obtained by dividing an integer by a non-negative power of ten, i.e., expressible as ***i*** / 10^{n} where ***i*** and ***n*** are integers and ***n*** ≥ 0. Precision is not reflected in this value space; the number 2.0 is not distinct from the number 2.00. (The datatype precisionDecimal may be used for values in which precision is significant.) The order relation on decimal is the order relation on real numbers, restricted to this subset.
Int	The precisionDecimal datatype represents the numeric value and (arithmetic) precision of decimal numbers which retain precision; it also includes values for positive and negative infinity and for "not a number", and it differentiates between "positive zero" and "negative zero". This datatype is introduced to provide a variant of decimal that closely corresponds to the floating-point decimal datatypes described by the expected forthcoming revision of IEEE/ANSI 754. Precision of values is retained and values are included for two zeroes, two infinities, and not-a-number.
Enumeration	Enumeration constrains the ·value space· to a specified set of values. Enumeration does not impose an order relation on the ·value space· it creates; the value of the ·ordered· property of the derived datatype remains that of the datatype from which it is derived.

2.3 Security

Purolator E-Ship Web Services authentication uses the standard HTTP Basic Authentication method allowing client programs to provide credentials in the form of a user name and password when making an HTTP request. Purolator E-Ship Web Services uses the standard HTTP protocol and in our case the username corresponds to the Development and Production Key that is issued to you by Purolator when you signed up for Purolator E-Ship Web Services. These credentials are sent from your client programs to our Purolator E-Ship Web Services servers securely using SSL.

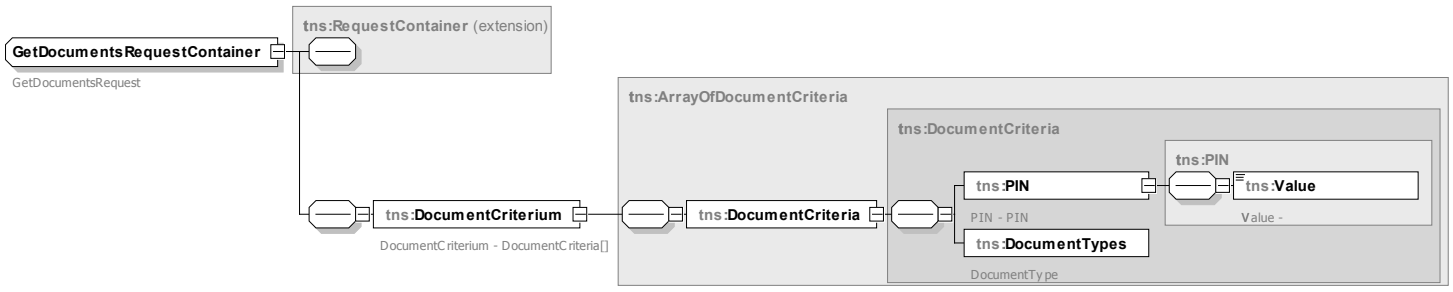
2.4 Service Methods and Objects

The following section will highlight the various methods associated with the shipment service. As well, visual representations will depict the XML schema of both the requests, and the responses of the service calls.

2.4.1 Get Documents

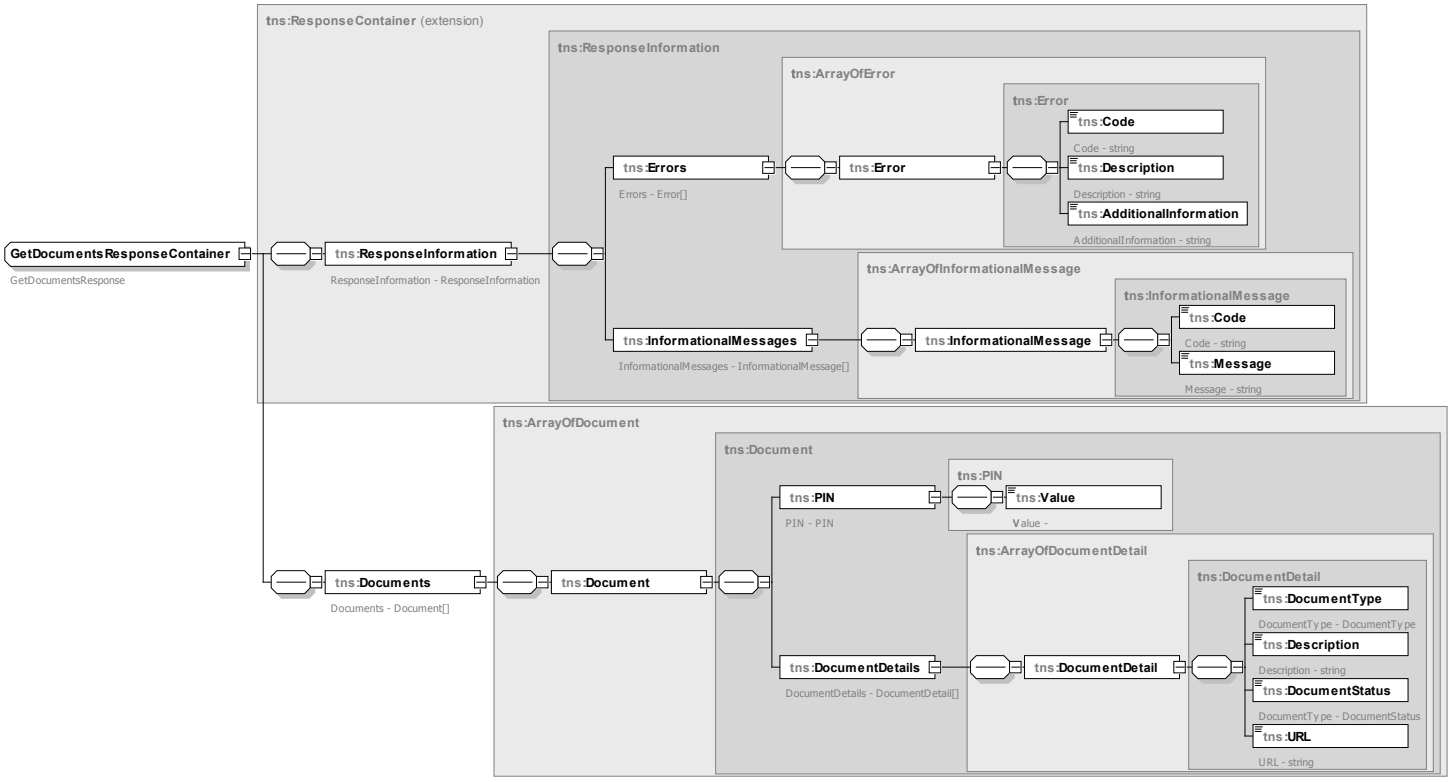
The getdocuments method is used to obtain shipment labels and/ or additional shipping documentation.
Note: The retention period for all shipping documents is same day. If you wish to reprint a shipping document, you must submit another GetDocuments request.

2.4.1.1 Request Diagrams – GetDocuments



Purolator
Where business is going

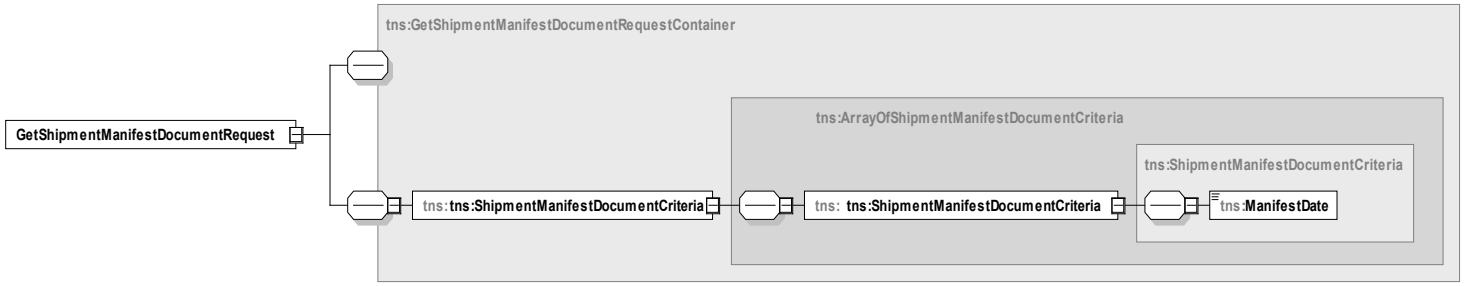
2.4.1.2 Response Diagrams – GetDocuments



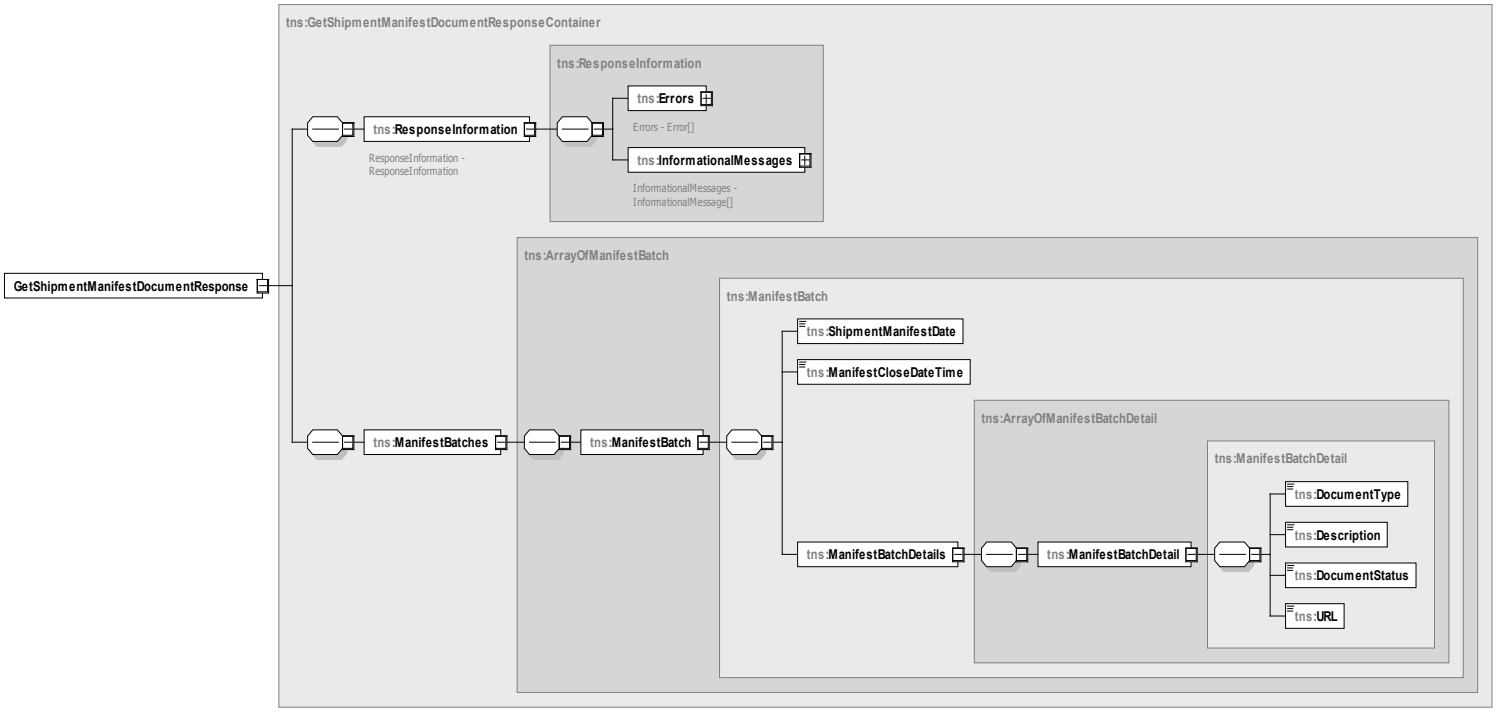
2.4.2 Get Shipment Manifest Document

The GetShipmentManifestDocument method is used to obtain the shipping manifest for the date requested.
Note: The retention period for all shipping documents is same day. If you wish to reprint a shipping document, you must submit another GetShipmentManifestDocument request. Manifests can be reprinted up to 24 months after consolidated date.

2.4.2.1 Request Diagrams – GetShipmentManifestDocument



2.4.2.2 Response Diagrams – GetShipmentManifestDocument

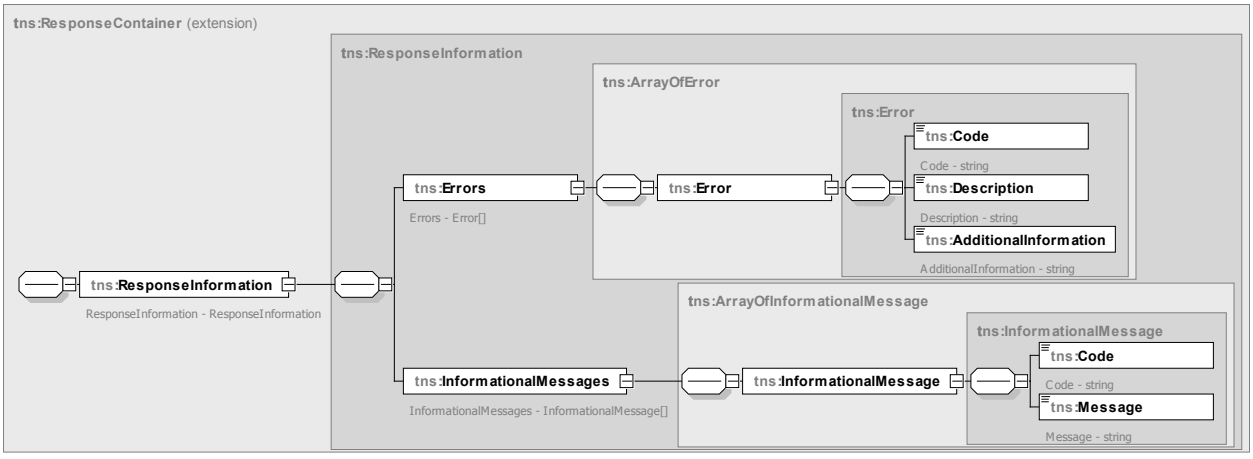


2.5 Error Handling

The Web Service calls return error data that your client application can use to identify and resolve runtime errors. If an error occurs during the invocation of most Web Services calls, then the API provides the following types of error handling:

- For errors resulting from badly formed messages, failed authentication, or similar problems, the Web Services returns a SOAP fault message.
- For errors resulting from a problem with the information sent to Purolator's systems, error codes and error descriptions are returned to the client application.

The following SOAP example shows the response to a request that contains an invalid postal code, and receiver phone number.



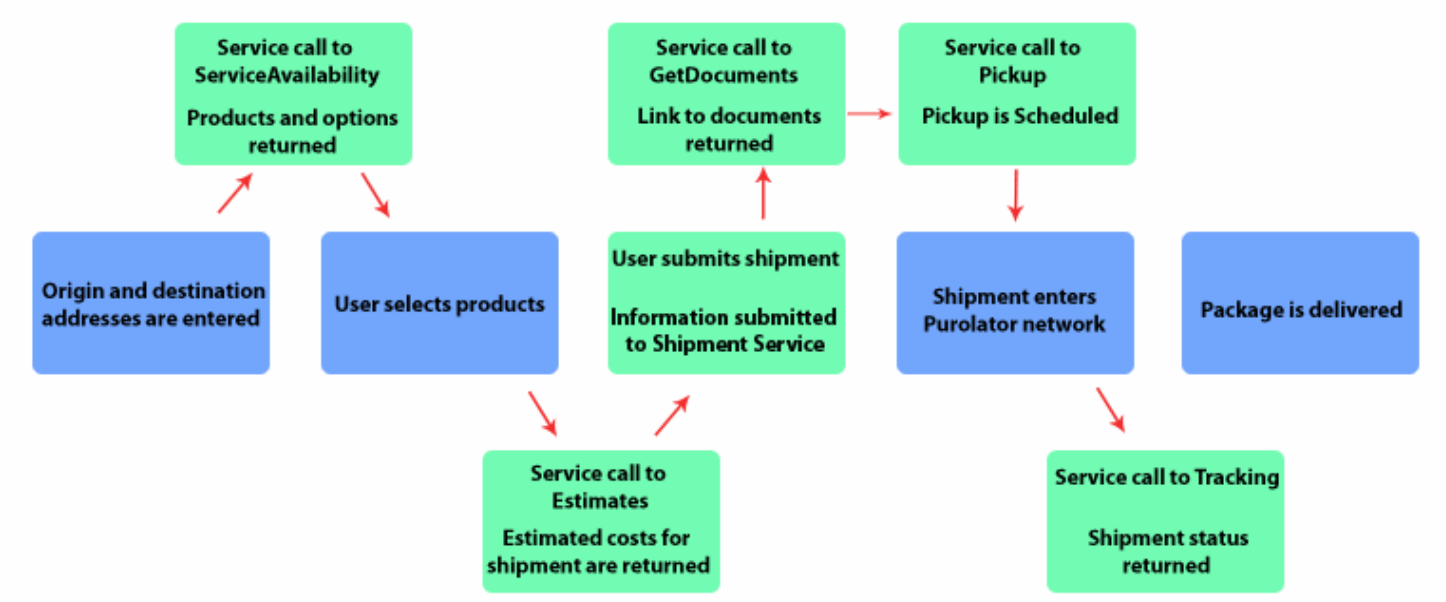
2.6 Implementation Considerations

As Purolator's business changes to meet the on-going needs of its customer, the business rules for certain products may change from time to time. As our business changes, your application doesn't have to. Purolator E-Ship Web Services offers customers an advanced Service Availability service that, when called, will return available products and services for a given origin and destination address. In addition to the available products and services, all business rules as well as product rules, product inclusions and exclusions, and products min/max dimensions and weights.

This can be used to offer your customers a greater experience, and offer all of Purolator's products and services, without needing to submit a request, only to discover that a particular product is not available for a particular address.

As well, additional Purolator E-Ship Web Services can be consumed to offer a richer customer experience. From estimates to tracking to retrieving necessary documentation, Purolator's E-Ship Web Services are designed to offer you the flexibility to consume the services required for your business.

Below is an example of how the all Purolator E-Ship Web Services may be utilized. (Green boxes denote service calls)



2.7 Error Messages / codes

Below is a list of all the error codes and descriptions.

Web Service Error

Code	Message
1000100	System exception occurs.
1100000	One or more errors have occurred.
1100100	Field is missing.
1100101	The Account Number is a mandatory field.

Code	Message
1100102	The Account Name is a mandatory field.
1100103	The Account Number is a mandatory field.
1100200	Field is invalid.
1100411	Account Status is invalid.
1100424	PIN %pin% is invalid.
1100505	Undefined Account Type for Account %account%.
1100676	Invalid Printer Type.
3001099	Invalid PackageType Code: {0}. Valid values are: {1}.
3001100	Invalid TaxType Code: {0}. Valid values are: {1}.
3001101	Invalid DataType Code: {0}. Valid values are: {1}.
3001102	Invalid ScanType Code: {0}. Valid values are: {1}.
3001103	Invalid SignatureImageFormat Code: {0}. Valid values are: {1}.
3001104	Invalid ILLanguage Code: {0}. Valid values are: {1}.
3001105	Invalid ESOLanguage Code: {0}. Valid values are: {1}.
3001106	Invalid ESOYesNo Code: {0}. Valid values are: {1}.
3001107	Invalid Option Code: {0}. Valid values are: {1}.
3001108	Invalid Product Code: {0}. Valid values are: {1}.
3001109	Invalid StreetType Code: {0}. Valid values are: {1}.
3001110	Invalid StreetDir Code: {0}. Valid values are: {1}.
3001111	Invalid StreetSuffix Code: {0}. Valid values are: {1}.
3001112	Invalid Country Code: {0}. Valid values are: {1}.
3001113	Invalid Province/State Code: {0}. Valid values are: {1}.

Code	Message
3001114	Translation Failed.
3001115	Service Call Failed.
3001116	Service Failed
3001117	A Web Service Error has occurred.
3001122	Invalid ChainOfSignature Code: {0}. Valid values are: {1}.
3001123	Invalid DangerousGoods Code: {0}. Valid values are: {1}.
3001124	Invalid DangerousGoodsClassification Code: {0}. Valid values are: {1}.
3001125	Invalid DangerousGoodsMode Code: {0}. Valid values are: {1}.
3001126	Invalid ExpressCheque Code: {0}. Valid values are: {1}.
3001127	Invalid ExpressChequeMethodOfPayment Code: {0}. Valid values are: {1}.
3001128	Invalid HoldForPickup Code: {0}. Valid values are: {1}.
3001129	Invalid SaturdayDelivery Code: {0}. Valid values are: {1}.
3001130	Invalid SaturdayPickup Code: {0}. Valid values are: {1}.
3001131	Invalid SpecialHanding Code: {0}. Valid values are: {1}.
3001132	Invalid TypesOfSpecialHandling Code: {0}. Valid values are: {1}.
3001133	Invalid DocumentType Code: {0}. Valid values are: {1}.
3001134	Invalid ExpressChequeAmount Code: {0}. Valid values are: {1}.
3001136	Service version is invalid
3001137	Invalid Declared Value Code: {0}. Valid values are: {1}.
3001138	Invalid OSNR Value Code: {0}. Valid values are: {1}.
3001139	Invalid registered shipping account specified in shipment
3001140	Invalid registered shipping account specified in return shipment

Code	Message
3001141	Postal / Zip code {0} code is not valid for Canada or U.S.
3001142	Country is not found
3001143	Suggested address is supplied
3001144	One or more addresses are invalid or suggestions are supplied
3001145	Invalid Piece Option Code: {0}. Valid values are: {1}.
3001146	Invalid Printer Type Option Code: {0}. Valid values are: {1}.
3001147	Invalid Address.
3001148	Postal / Zip Code is required to validate a Canadian / U.S. Address.
3001153	Service Error

APPENDIX A – Complete Field List

ArrayOfDocument | ArrayOfDocumentCriteria | ArrayOfDocumentDetail | ArrayOfError | ArrayOfInformationalMessage | Document | DocumentCriteria | DocumentDetail | DocumentStatus | DocumentType | Error | GetDocumentsRequestContainer | GetDocumentsResponseContainer | GetManifestDocumentRequest | GetManifestDocumentResponse | InformationalMessage | Language | ManifestBatch | ManifestBatchDetails | PIN | RequestContext | ResponseContext | ResponseInformation | ShipmentManifestDocumentCriteria

Input

Field Name	Description	Output Values
ComplexType RequestContext	**Used by: RequestContext**	
Version	Version Number of the Web Service Request.	1.0 or 1.1(Currently)
Language	Used to define the Language of the Response text.	Enumeration. en fr
GroupID	For Future Use	
RequestReference	Reference Identifier for the Service Request. Used to aid in Customer Support.	String
ComplexType GetDocumentsRequestContainer		
DocumentCriterium		**Complex Type DocumentCriterium**
ComplexType DocumentCriterium	**Used by: GetDocumentsRequestContainer**	
ArrayOfDocumentCriteria		**Complex Type ArrayOfDocumentCriteria**
ComplexType ArrayOfDocumentCriteria	**Used by: GetDocumentsRequestContainer/DocumentCriterium**	
DocumentCriteria		**Complex Type DocumentCriteria**
ComplexType DocumentCriteria	**Used by: GetDocumentsRequestContainer/DocumentCriterium**	
PIN	PIN number for document request.	String Value
DocumentTypes	Array of document types	**Complex Type DocumentType**
ComplexType DocumentType	**Used by: GetDocumentsRequestContainer/DocumentCriterium**	

Input

Field Name	Description	Output Values
DocumentType		**Enumeration:** COSBillOfLading CustomsInvoice CustomsInvoiceThermal DangerousGoodsDeclaration DomesticBillOfLading DomesticBillOfLadingThermal ExpressChequeReceipt ExpressChequeReceiptThermal FCC740 FDA2877 InternationalBillOfLading InternationalBillOfLadingThermal NAFTA
ComplexType GetShipmentManifestDocumentRequest	**Used by: GetShipmentManifestDocumentRequestContainer**	
ShipmentManifestDocumentCriterium	Array to submit document request.	**Complex type ShipmentManifestDocumentCriteria**
ComplexType ShipmentManifestDocumentCriteria	**Used by: GetShipmentManifestDocumentRequest**	
ManifestDate	Required. Date of the shipping date.	Date. Format: YYYY-MM-DD

Output

Field Name	Description	Output Values
Element GetDocumentsResponseContainer		
ResponseInformation **Documents**		**Complex Type ResponseInformation** **Complex Type ArrayOfDocuments**
Element ResponseInformation	**Used by: ResponseInformation**	
ResponseReference **Errors** **InformationalMessages**	Array of errors	**Complex Type ArrayOfError** **Complex Type ArrayOfInformationalMessages**
ComplexType ArrayOfError	**Used by: ArrayOfError, ResponseInformation/Errors**	
Error	Error information returned by the service call.	**Complex Type Error**
ComplexType Error	**Used by: Error, ArrayOfError/Error**	

Output

Field Name	Description	Output Values
Code	Error Code	
Description	Error Description	
AdditionalInformation	Additional error information	
ComplexType ArrayOfInformationalMessage	**Used by: ArrayOfInformationalMessage, ResponseInformation/InformationalMessages**	
InformationalMessage		
Element ArrayOfPIN	**Used by: ArrayOfInformationalMessage, ResponseInformation/InformationalMessages**	
PIN	Array of PIN	
Element ArrayOfDocument	**Used by: GetDocumentsResponseContainer/Documents**	
Document		**Complex Type Document**
Element Document	**Used by: GetDocumentsResponseContainer/Documents**	
PIN		**Complex Type PIN**
DocumentDetails		**Complex Type ArrayOfDocumentDetail**
Element PIN	**Used by: GetDocumentsResponseContainer/Documents**	
Value	PIN number	String
Element ArrayOfDocumentDetail	**Used by: GetDocumentsResponseContainer/Documents**	
DocumentDetail		**Complex Type DocumentDetail**
Element DocumentDetail	**Used by: GetDocumentsResponseContainer/Documents**	
DocumentType		**Complex Type DocumentType**
Description	Description of document type	String
DocumentStatus	Status of document	**Enumeration:** Pending Queued Processing Completed Error
URL	URL of Documents	String
ComplexType ResponseContext		
ResponseReference		

Output

Field Name	Description	Output Values
Element GetShipmentManifestDocument Response	**Used by: GetShipmentManifestDocument**	
ResponseInformation		**Complex Type ResponseInformation**
ManifestBatches		**Complex Type ManifestBatches**
Element ManifestBatches	**Used by: GetDocumentsResponseContainer/Documents**	
ManifestBatch		**Complex Type ManifestBatch**
Element ManifestBatch	**Used by: GetDocumentsResponseContainer/Documents**	
ShipmentManifestDate		
ManifestCloseDateTime		
ManifestBatchDetails		**Complex Type ArrayOfManifestBatchDetails**
Element ArrayOfManifestBatchDetails	**Used by: GetDocumentsResponseContainer/Documents**	
DocumentType		
Description		
DocumentStatus		
URL		