eShipper API Document

Version 4.1.0

Table of Contents

1. Introduction

This document outlines the details of the API specifications for integrating with the EShipper application. The EShipper API is a work in progress, and it is expected that some minor changes may be made to the API before a fully stable version is released periodically

The protocol of communication between the client and the EShipper application will be based on XML technology. Currently, the EShipper will handle 6 types of requests from the client, (i) Quote Request, (ii) Ship Request, (iii) Order Information Request, (iv) Cancel Order Request, (v) Invoice Search Request, and (vi) Invoice Request. All requests will be sent via the client in XML format, and the EShipper application will respond to these requests in XML format.

2. Set Up Process

Any customer willing to use the XML API will require to be registered with the application first. Please contact EShipper administrator or your sales representative to ensure that you have been set-up on the application. The Customer can fetch their API Username and Password by logging in to their eShipper Account and clicking on "My Account". This API Username and Password will be sent in every API request sent to eShipper for authorization. The EShipper application will use the customer name and password to authenticate the request before processing it.

All PRODUCTION requests will be sent to the URL https://web.eshipper.com/rpc2

Customers can also use the eShipper's TEST server to POST test requests. The URL for the test server is:

https://test.eshipper.com/rpc2

3. XML Schema

The XML schema for the EShipper API is shown below. This schema contains definitions for elements and attributes used in the 8 message types:- (i) QuoteRequest, (ii) QuoteReply, (iii) ShippingRequest, (iv) ShippingReply, (v) ShipmentCancelrequest, (vi) ShipmentCancelReply, (vii) OrderInformationRequest, (viii) OrderInformationReply, (ix) BoxingRequest, and (x) BoxingReply

1. Version Management (since v3.0.0)

The use of version in the root element has been introduced as of v3.0.0. For all requests, version information will have to be sent in the <EShipper> element

Example of version in the EShipper element: -

```
<Eshipper xmlns="http://www.eshipper.net/XMLSchema"
username="merchant" password="9999" version="3.0.0">
```

4. Getting a Quote

In order to receive a quote from the EShipper API, the client can send a “QuoteRequest” message to the API. The API will process the request and return a “QuoteReply” message. The elements and attributes required by the “Quote” request and response messages are shown in the schema above. A sample request and response is shown below:-

Available Tags with Attributes in the QuoteRequest

- QuoteRequest [Required]
    - serviceId [Optional]
    - dangerousGoodsType [Optional]
    - isSaturdayService [Optional]
    - signatureRequired [Optional]
    - holdForPickupRequired [Optional]
    - specialEquipment [Optional]
    - deliveryAppointment [Optional]
    - insideDelivery [Optional]
    - scheduledShipDate [Optional]
    - insuranceType [Optional]
    - insidePickup [Optional]
    - saturdayPickupRequired [Optional]
    - singleShipment [Optional]
    - crossBorderFee [Optional]
    - excessLength [Optional]
    - limitedAccess [Optional]
    - customsIn-bondFreight [Optional]
    - militaryBaseDelivery [Optional]
    - exhibitionConventionSite [Optional]
    - pierCharge [Optional]
    - homelandSecurity [Optional]
    - stackable [Optional]
    - Sort and Segregate Charge [Optional]

- From [Required]
    - id [Required]
    - company [Required]
    - address1 [Required]
    - address2 [Optional]
    - city [Required]
    - state [Required]
    - country [Required]
    - zip [Required]
    - residential [[Optional]
    - tailgateRequired [Optional]
    - phone [Optional]
    - attention [Optional]
    - email [Optional]
    - instructions [Optional]
    - confirmDelivery [Optional]
- To [Required]
    - id [Required]
    - company [Required]
    - address1 [Required]
    - address2 [Optional]
    - city [Required]
    - state [Required]
    - country [Required]
    - zip [Required]
    - residential [[Optional]
    - tailgateRequired [Optional]
    - phone [Optional]
    - attention [Optional]
    - email [Optional]
    - instructions [Optional]
    - notifyRecipient [Optional]
- COD [Optional]
    - paymentType [Required]
- CODReturnAddress [Required if COD is selected]
    - codCompany [Required]
    - codName [Required]
    - codAddress1 [Required]
    - codCity [Required]
    - codStateCode [Required]
    - codZip [Required]
    - codCountry [Required]
- Packages [Required]
    - type [Required][valid package types are:Envelope,Courier Pak,Package and Pallet]
- Package [Required]
    - length [Required]
    - width [Required]
    - height [Required]
    - weight [Required]
    - weightOz [Required for DHLeC Shipments, >= zero]
    - type [Optional][Required if type="Pallet"]freightClass

    [Optional] [Required if type=“Pallet”]
    - nmfcCode [Optional] [Required if type=“Pallet”]
    - insuranceAmount [Optional]
    - codAmount [Optional]
    - description [Optional][Required if type=“Pallet”]
- Pickup [Optional]
    - contactName [Required]
    - phoneNumber [Required]
    - pickupDate [Required]
    - pickupTime [Required]
    - closingTime [Required]
    - location [Optional]

Sample QuoteRequest message

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

<EShipper xmlns="http://www.eshipper.net/XMLSchema" username="merchant
inc." password="9999" version="3.0.0">

      <QuoteRequest serviceId="0" stackable="true" >

            <From id="123" company="Test Company" address1="650 CIT
            Drive" city="Livingston" state="ON" country="CA"
            zip="L4J7Y9" />


            <To company="Test Company" address1="650 CIT Drive"

            city="Livingston" state="MA" zip="01603" country="CA" />


            <!-- COD is optional -->

            <COD paymentType="Check">

                  <CODReturnAddress codCompany="ABC Towing"
                  codName="Alfred" codAddress1="444 Highway 401"
                  codCity="Toronto" codStateCode="ON" codZip="A1B2C3"
                  codCountry="CA"/>

            </COD>


            <!-- If type="Envelope" or type="Courier Pak" or
            type=”Package” package information is required -->

            <Packages type="Package">
                  <!-- Description is ignore for quote requests -->

                  <!-- If package type="Pallet", type and freightClass
                  are required for quote requests -->
            <Package length="15" width="10" height="12"
            weight="10" type="Pallet" freightClass="70"
            nmfcCode="123456"   insuranceAmount="0.0"
            codAmount="0.0" description="desc."/>

            <Package length="15" width="10" height="10"
            weight="5" type="Pallet" freightClass="70"
            insuranceAmount="0.0" codAmount="0.0"
            description="desc."/>

        </Packages>


        <!-- Pickup is optional -->

        <Pickup contactName="Test Name" phoneNumber="888-888-8888"
        pickupDate="2009-08-03" pickupTime="16:30"
        closingTime="17:45" location="Front Door"/>

    </QuoteRequest>

</EShipper>
```

Sample QuoteReply response to QuoteRequest message

```
<EShipper xmlns="http://www.eshipper.net/XMLSchema" version="3.0.0">

    <QuoteReply>

        <Quote carrierId="2" carrierName="Purolator" serviceId="4"
        serviceName="Air" modeTransport="A" transitDays="1"
        baseCharge="177.0" fuelSurcharge="0.0" totalCharge="177.0"
        currency="CAD">

        </quote>


        <Quote carrierId="2" carrierName="Purolator" serviceId="13"
        serviceName="Ground" modeTransport="G" transitDays="1"
```

```
                baseCharge="28.650000000000002" fuelSurcharge="0.0"
                totalCharge="28.65" currency="CAD">

                </quote>



                <Quote carrierId="1" carrierName="Federal Express"
                serviceId="1" serviceName="Priority" modeTransport="null"
                transitDays="0" baseCharge="46.27000045776367"
                fuelSurcharge="6.25" totalCharge="52.52" currency="CAD">

                </quote>



                <Quote carrierId="1" carrierName="Federal Express"
                serviceId="3" serviceName="Ground" modeTransport="null"
                transitDays="0" baseCharge="30.739999771118164"
                fuelSurcharge="0.0" totalCharge="31.82" currency="CAD">

                      <Surcharge id="null" name="Other"
                      amount="1.0800000429153442"/>

                </quote>



                <Quote carrierId="3" carrierName="Canada WorldWide"
                serviceId="16" serviceName="Air Freight"
                modeTransport="null" transitDays="0" baseCharge="300.0"
                fuelSurcharge="36.0" totalCharge="336.0" currency="CAD">

                </quote>



                <Quote carrierId="3" carrierName="Canada WorldWide"
                serviceId="15" serviceName="Next Flight Out"
                modeTransport="null" transitDays="0" baseCharge="165.0"
                fuelSurcharge="19.8" totalCharge="184.8" currency="CAD">

                </quote>


          </QuoteReply>

    <Eshipper>
```

4.1 Currency Code (since v3.0.0)

The API will return currency code attribute in the Quote response. This attribute will be return in the Quote element.

Example of currency code in the Quote element:-

```
<Quote carrierId="3" carrierName="Canada WorldWide"
serviceId="15" serviceName="Next Flight Out"
modeTransport="null" transitDays="0" baseCharge="165.0"
fuelSurcharge="19.8" totalCharge="184.8" currency="CAD">

</quote>
```

5. Shipping an Order

In order to ship an order using the EShipper API, the client can send a "ShippingRequest" message to the API. The application then attempts to ship the order using the information provided. If a serviceId is specified in the request, the application will attempt to ship the order using that service only. If no serviceId attribute is specified, the application will retrieve rates for all eligible services and will ship using the cheapest fastest service available, i.e. it will first filter the services by the fastest and then pick the cheapest amongst these.

Available Tags with Attributes in the ShippingRequest

- ShippingRequest [Required]
  - serviceId [Optional]
  - dangerousGoodsType [Optional]
  - isSaturdayService [Optional]
  - signatureRequired [Optional]
  - holdForPickupRequired [Optional]
  - specialEquipment [Optional]
  - deliveryAppointment [Optional]
  - insideDelivery [Optional]
  - scheduledShipDate [Optional]
  - insuranceType [Optional]
  - insidePickup [Optional]
  - singleShipment [Optional]
  - crossBorderFee [Optional]
  - excessLength [Optional]
  - limitedAccess [Optional]
  - customsIn-bondFreight [Optional]
  - militaryBaseDelivery [Optional]
  - exhibitionConventionSite [Optional]

  - pierCharge [Optional]
  - homelandSecurity [Optional]
  - readyTimeHrs [Optional]
  - readyTimeMin [Optional]
  - originCloseTimeHrs [Optional]
  - originCloseTimeMin [Optional]
  - destinationCloseHrs[Optional]
  - destinationCloseMin [Optional]
  - stackable [Optional]
  - Sort and Segregate Charge [Optional]
- From [Required]
  - id [Required]
  - company [Required]
  - address1 [Required]
  - address2 [Optional]
  - city [Required]
  - state [Required]
  - country [Required]
  - zip [Required]
  - residential [[Optional]
  - tailgateRequired [Optional]
  - phone [Required]
  - attention [Required]
  - email [Optional]
  - instructions [Optional]
  - confirmDelivery [Optional]
- To [Required]
  - id [Required]
  - company [Required]
  - address1 [Required]
  - address2 [Optional]
  - city [Required]
  - state [Required]
  - country [Required]
  - zip [Required]
  - residential [[Optional]
  - tailgateRequired [Optional]
  - phone [Required]
  - attention [Required]
  - email [Optional]
  - instructions [Optional]
  - notifyRecipient [Optional]
- Shipper [Optional – Only for In-bond Manifest Customers]
  - id [Required]
  - company [Required]
  - address1 [Required]
  - address2 [Optional]
  - city [Required]
  - state [Required]
  - country [Required]
  - zip [Required]
  - residential [[Optional]

  - tailgateRequired [Optional]
  - phone [Required]
  - attention [Required]
  - email [Required]
  - instructions [Optional]
  - notifyRecipient [Optional]
- Payment [Required]
  - type [Required]
- Reference [Optional]
  - name [Required]
  - code [Required]
- COD [Optional]
  - paymentType [Required]
- CODReturnAddress [Required if COD is selected]
  - codCompany [Required]
  - codName [Required]
  - codAddress1 [Required]
  - codCity [Required]
  - codStateCode [Required]
  - codZip [Required]
  - codCountry [Required]
- Packages [Required]
  - type [Required][valid package types are:Envelope,Courier Pak,Package and Pallet]
- Package [Required]
  - length [Required]
  - width [Required]
  - height [Required]
  - weight [Required]
  - weightOz [Required for DHLeC Shipments, >= zero]
  - type [Optional][Required if type="Pallet"]
  - freightClass [Optional] [Required if type="Pallet"]
  - nmfcCode [Optional] [Required if type="Pallet"]
  - insuranceAmount [Optional]
  - codAmount [Optional]
  - description [Optional][Required if type="Pallet"]
- Pickup [Optional]
  - contactName [Required]
  - phoneNumber [Required]
  - pickupDate [Required]
  - pickupTime [Required]
  - closingTime [Required]
  - location [Optional]
- PackingSlip [Optional]
  - PackingItem [Required]
    - name [Required]
    - sku [Required]
    - quantity [Required]
    - weight [Required]
- CustomsInvoice [Optional] [Required if International shipment.]
  - Currency [optional]

- brokerName [optional]
- shipperTaxID [optional]
- receiverTaxID [optional]
- contactCompany [optional]
- contactName [optional]
- contactPhone [optional]
- BillTo [Required]
  - company [Required]
  - name [Required]
  - address1 [Required]
  - city [Required]
  - state [Required]
  - zip [Required]
  - country [Required]
- Contact [Required]
  - name [Required]
  - phone [Required]
- Item [Required]
  - code [Required]
  - description [Required]
  - originCountry [Required]
  - quantity [Required]
  - unitPrice [Required]
  - skuCode [Required for DHLeC Shipments]
- DutiesTaxes [Optional]
  - billTo [optional][values=’Receiver’, ‘Shipper’, ‘Consignee Account’][default=’Receiver’]
  - dutiable [Optional][values=’No’,’Yes’] [default=’No’]
  - consigneeAccount [Required if billTo=’Consignee Account’]
  - sedNumber [Optional]
- In-Bond Manifest Information [Optional]
  - locationOfGoods [optional]
  - Carrier [optional]
  - VehicleId [optional]
  - CustomsClearedBy [optional]
  - HandlingInfo [optional]
  - PreviousCargoNum [optional]
  - Weight [optional]
  - WeightUOM [optional; KGS or LBS]

Sample ShippingRequest message

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

<EShipper xmlns="http://www.eshipper.net/XMLSchema" username="merchant
inc." password="abcd" version="3.0.0">

     <ShippingRequest serviceId="3" stackable="true" >

          <From id="123" company="Test Company" address1="650 CIT
          Drive" city="Livingston" state="ON" zip="L8E5X9"
```

```
country="CA" phone="9052223333" attention="Riz"
email="riz@shaw.ca" residential="true" />


<To company="Test Company" address1="650 CIT Drive"
city="Livingston" state="BC" zip="V3N4R3" country="CA"
phone="4162223333" attention="RizTo" email="riz@shaw.ca"/>


<!-- COD is optional -->

<COD paymentType="Check">

      <CODReturnAddress codCompany="ABC Towing"
      codName="Alfred" codAddress1="444 Highway 401"
      codCity="Toronto" codStateCode="On" codZip="A1B2C3"
      codCountry="CA"/>

</COD>


<!-- If type="Envelope" or type="Courier Pak" or
type=”Package” or type=”Pallet” package information is
required -->

<Packages type="Package">


      <!-- If package type="Pallet", type and freightClass
      are required for shipping requests -->

      <Package length="15" width="10" height="12"
      weight="12" type="Pallet" freightClass="70"
      insuranceAmount="0.0" codAmount="0.0"
      description="desc."/>

      <Package length="15" width="10" height="10"
      weight="14" type="Pallet" freightClass="70"
      nmfcCode="123456" insuranceAmount="0.0"
      codAmount="0.0" description="desc."/>

</Packages>


<!-- Pickup is optional -->
```

```
<Pickup contactName="Test Name" phoneNumber="888-888-8888"
pickupDate="2009-08-03" pickupTime="16:30"
closingTime="17:45" location="Front Door"/>


<Payment type="3rd Party" />


<!-- References are optional and up to 3 may be specified.
First reference specified is primary reference, second
reference is secondary reference and so on-->

<Reference name="" code="123456" />

<Reference name="" code="" />

<Reference name="" code="" />

<!--PackingSlip is optional, required if Packing Slip
needs to be printed-->


<PackingSlip>

      <PackingSlip name="TestProd1" sku="2234"
      quantity="1" weight="12" />

      <PackingSlip name="TestProd2" sku="4234"
      quantity="5" weight="3" />

</PackingSlip>


<!--CustomsInvoice is optional, required for international
shipment-->

<CustomsInvoice brokerName=”John Doe”
contactCompany=”MERITCON INC” contactName=”Jim”>

      <BillTo company="ABC Towing" name="Alfred"
      address1="444 Highway 401" city="Toronto" state="ON"
      zip="A1B 2C3" country="CA" />


      <Contact name="Rizwan" phone="555-555-4444" />


      <Item code="1234" description="Laptop computer"
      originCountry="US" quantity="100" unitPrice="1000.00"
      />
```

```
            <DutiesTaxes dutiable="true" billTo="receiver" />


            <InBondManifest locationOfGoods="CANADA WORLDWIDE
            SLC4358" nameOfCarrier="AIR CANADA"
            vehicleIdentification="12345"
            customsClearedBy="CANADA WORLDWIDE"
            handlingInfo="CANADA WORLDWIDE SERVICES INC. TO
            ARRANGE DELIVERY" previousCargoControlNum="1234"
            weight="12.34" weightUOM="LBS"/>


        </CustomsInvoice>

    </ShippingRequest>

</EShipper>
```

Sample ShippingReply response to ShippingRequest message

```
<EShipper xmlns="http://www.eshipper.net/xml/XMLSchema" version="3.0.0">

    <ShippingReply>

        <Order id="181004" />

        <Carrier carrierName="Federal Express" serviceName="FedEx
        Ground" />

        <Reference code="1234567" name="RizReference" />

        <Package trackingNumber="052800410000484" />

        <Package trackingNumber="052800410000491" />


        <Pickup confirmationNumber ="123456789" />


        <TrackingURL>http://www.fedex.com/Tracking?tracknumbers=052
        800410000484</TrackingURL>

        <Labels>[base-64 encoded String]</Labels>

        <CustomsInvoice>[base-64 encoded String]</CustomsInvoice>

        <PackingSlip>[base-64 encoded String]</PackingSlip>
        <!--There will be only one quote element in the shipping
        reply-->

        <Quote carrierId="1" carrierName="Federal Express"
        serviceId="3" serviceName="Ground" modeTransport="null"
        transitDays="0" baseCharge="30.739999771118164"
        fuelSurcharge="0.0" totalCharge="31.82" currency="CAD">


            <Surcharge id="null" name="Other"
            amount="1.0800000429153442"/>

        </quote>

    </ShippingReply>

</EShipper>
```

As seen in the request above, the serviceId was set to “3” which implies that a “FedEx Ground” shipment was requested (see section on “Services” for more information on this feature). The application performed the shipment and returned the tracking numbers for the packages along with the carrier and service information.

1. Labels

The labels are returned in Acrobat PDF format as a base-64 encoded string inside the <Labels> tag. In order to recreate the labels on the client side, the client will have to read the characters enclosed inside this “Labels” tag, decode it using a base-64 decoding algorithm (which is readily available on most platforms), and then create an Acrobat PDF file from the resulting bytes. Please see appendix A for some examples on this process.

2. Customs Invoice

If customs invoice information is submitted in the shipping request, it is returned in Acrobat PDF format as a base-64 encoded string inside the <CustomsInvoice> tag. In order to recreate the customs invoice on the client side, the client will have to read the characters enclosed inside this “CustomsInvoice” tag, decode it using a base-64 decoding algorithm (which is readily available on most platforms), and then create an Acrobat PDF file from the resulting bytes. Please see appendix A for some examples on this process.

In-bond Manifest Section: This section is to be sent only if you are set up as an in-bond manifest customer. By default, customers are not set up as this type of customer and in most cases you are not required to send this information. Also, the ‘Shipper’ element is needed only if you are an in-bond manifest customer.

3. Tracking URL (since v2.0.4)

The API will return a tracking URL in the ShippingReply response. This URL can be used to track the shipment on the carrier’s website. If no tracking URL is available, then the API will return an empty string.

4. Label Images (since v2.0.7)

The API returns label images if the images are available from the carrier. Currently, the API returns PNG images only. For multi-package shipments the API will return one label image per package. The images are sent under the <LabelData/> element. Each label will have its own <Label> element, containing sub-elements <Type> and <Data>. The image data is contained in the <Data> element as a base-64 encoded string (similar to the PDF data). The contents of the <Data> element will have to be decoded using Base-64 decoding algorithm.

For example, a shipment with 2 packages will return images in the following method:-

```
<LabelData>
      <Label>
            <Type>PNG</Type>
            <Data>[base-64 encoded String]</Data>
      </Label>
      <Label>
            <Type>PNG</Type>
            <Data>[base-64 encoded String]</Data>
      </Label>
</LabelData>
```

5. Quote (since v3.0.0)

The API will return a Quote element in the ShippingReply response. The Quote element contains the rate information about the shipment.

6. Pickup confirmation number (since v3.0.0)

The API will return a Pickup element in the ShippingReply response. The Pickup element contains the pickup confirmation number of the pickup request. The attribute confirmationNumber is optional.

```
<Pickup confirmationNumber ="123456789" />
```

6. Shipment Cancel Request

In order to cancel an order that was previously created using the EShipper API, the client can send a “ShipmentCancelRequest” message to the API. The system will attempt to cancel the order based on the ‘orderId’ or ‘trackingNumber’ provided in the request. A success or error message will be returned depending on the result of the request. A status of ‘4’ indicates that the shipment was cancelled. For a list of status mappings, please see Appendix B, section 10.1.

Sample ShipmentCancelRequest message with Order ID

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

<es:EShipper xmlns:es="http://www.eshipper.net/XMLSchema"
username="test" password="test" version="3.0.0">

      <ShipmentCancelRequest>

            <Order orderId="383363"/>

      </ShipmentCancelRequest>

</es:EShipper>
```

Sample ShipmentCancelRequest message with Tracking Number

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

<es:EShipper xmlns:es="http://www.eshipper.net/XMLSchema"
username="test" password="test" version="3.0.0">

      <ShipmentCancelRequest>

            <Order trackingId="1234567890"/>

      </ShipmentCancelRequest>

</es:EShipper>
```

Sample ShipmentCancelReply response to ShipmentCancelRequest message

```
<EShipper xmlns="http://www.eshipper.net/xml/XMLSchema" version="3.0.0">

      <ShipmentCancelReply>

            <Order orderId="383363" message="Order has been cancelled!"
            />

            <Status statusId="4" />

      </ShipmentCancelReply>

</EShipper>
```

Note: Multiple Orders can be cancelled with one request by sending Order IDs separated by Comma.

```
        <ShipmentCancelRequest>

                <Order orderId="383363, 383364, 383365, 383366, 383367"/>

        </ShipmentCancelRequest>
```

Or by sending multiple Tracking Numbers separated by Comma, as:

```
        <ShipmentCancelRequest>

                <Order trackingId="1234567890, 3021110002, 5671212120"/>

        </ShipmentCancelRequest>
```

7. Order Information Request

Information on orders that were previously created can be requested using the 'OrderInformationRequest'. This request will return all the fields from the 'ShippingReply', plus a status element <Status> containing the status Id and corresponding status name for the order. For a list of status mappings, please see Appendix B, section 10.1.

Sample OrderInformationRequest message for version 3.0.0

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

<es:EShipper xmlns:es="http://www.eshipper.net/XMLSchema"
username="test" password="test" version="3.0.0">

        <OrderInformationRequest>

                <Order orderId="232212"/>

        </OrderInformationRequest>

</es:EShipper>
```

Sample OrderInformationRequest message for version 3.1.0

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

<es:EShipper xmlns:es="http://www.eshipper.net/XMLSchema"
username="test" password="test" version="3.1.0">

        <OrderInformationRequest>

                <Order orderId="232212" detailed="true" />

        </OrderInformationRequest>

</es:EShipper>
```

Sample OrderInformationRequest message with Reference-Code

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

<es:EShipper xmlns:es="http://www.eshipper.net/XMLSchema"
username="test" password="test" version="3.1.0">

      <OrderInformationRequest>

            <Order orderId="111" detailed="true" referenceCode="#56781" />

      </OrderInformationRequest>

</es:EShipper>
```

Sample OrderInformationReply response to OrderInformationRequest message for version 3.0.0

```
<EShipper xmlns="http://www.eshipper.net/xml/XMLSchema" version="3.0.0">

      <OrderInformationReply>

            <Order id="383361" />

            <Carrier carrierName="Purolator" serviceName="Purolator
            Ground" />

            <Reference code="123456" name="" />

            <Package trackingNumber="VVV000197197" />

            <Package trackingNumber="VVV000197198" />


            <Pickup confirmationNumber ="123456789" />


            <TrackingURL/>

            <Labels>[base-64 encoded String]</Labels>

            <Status statusId="3" statusName="DELIVERED" />


            <OriginalCharge carrierId="1" carrierName="Federal Express"
            serviceId="3" serviceName="Ground" modeTransport="null"
            transitDays="0" baseCharge="30.739999771118164"
            fuelSurcharge="0.0" totalCharge="31.82" currency="CAD"
            actualWeight="10">
                  <Surcharge id="null" name="Other"
                  amount="1.0800000429153442"/>

            </OriginalCharge>




            <ApplicableCharge carrierId="1" carrierName="Federal
            Express" serviceId="3" serviceName="Ground"
            modeTransport="null" transitDays="0"
            baseCharge="30.739999771118164" fuelSurcharge="0.0"
            totalCharge="31.82" currency="CAD" billedWeight="10">

                  <Surcharge id="null" name="Other"
                  amount="1.0800000429153442"/>

            </ApplicableCharge>

      </OrderInformationReply>
```

```
</Eshipper>
```

Sample OrderInformationReply response to OrderInformationRequest message for version 3.1.0

```
<EShipper xmlns="http://www.eshipper.net/xml/XMLSchema"
version="3.1.0">

<OrderInformationReply>

<Order id="949986" />

<Carrier carrierName="CANADA WORLDWIDE INC." serviceName="Canada
Worldwide Air Freight" />

<Reference code="" name="" />

<Package trackingNumber="0-1" />

<TrackingURL/>

<Pickup />

<Labels>[base-64 encoded String]</Labels>

<Status statusId="1" statusName="READY FOR SHIPPING" />

<Quote carrierId="3" carrierName="CANADA WORLDWIDE INC." serviceId="16"
serviceName="Canada Worldwide Air Freight" modeTransport=""
transitDays="0" currency="CAD" actualWeight="4" baseCharge="264.0"
fuelSurcharge="31.68000030517578" totalCharge="314.1390003051758">

<Surcharge id="711564" name="Insurance" amount="3.5" />

<Surcharge id="711565" name="GST" amount="14.959000000000001" />

</Quote>

<OrderDetails>

      <PickupFrom Company="test" Id="2745215" Address1="My address"
      Address2="pune" City="THORNHILL" Province="ON" Postal="L4J7Y9"
      Country="CA" Residential ="N" Phone="8888888888"
      Email="test@test.co" Attn="mani" Instructions="Inst 1" />

      <ShipTo Company="SumaSoft2" Id="2745214" Address1="My address"
      Address2="35" City="THORNHILL" Province="ON" Postal="L4J7Y9"
      Country="CA" Residential ="N" Phone="8888888888"
      Email="test@test.co" Attn="Mani" Instructions="Inst 2" />
```

```
        <Packaging Type="Packaging">

              <Package Length="3" Width="2" Height="1" Weight="4"
        Description="Item 1 Desc" InsuranceAmount="$500.00"/>

        </Packaging>

        <Quantity Value="1" />

        <Weight Value="4 lbs / 4 lbs" />

        <Insurance Value="$500.00" />

        <CarrierTracking No="0-1" />

        <ShipmentDate Date="2010-03-29 00:00:00.0" />

        <OrderStatusHistory>

              <Status Name="READY FOR SHIPPING" Date="2010-03-29
              14:35:10.0" AssignedBy="SYSTEM" Comments=""/>

        </OrderStatusHistory>

  </OrderDetails>

  </OrderInformationReply>

  </EShipper>
```

1. Original Charge and Applicable Charge (since v3.0.0)

The OriginalCharge element contains quoted rate information when shipment is created.

The ApplicableCharge element contains actual rate information after the final charges have been determined for the shipment (based on information received from carrier after shipment is delivered). The OrderInformationReply element may contain more than one ApplicableCharge element.

Once final charges have been determined, the 'OriginalCharge' element and 'ApplicableCharge' element(s) will be return inside the OrderInformationReply element.

If final charges have not yet been determined, the 'Quote" element will be return inside the OrderInformationReply element.

2. Pickup confirmation number (since v3.0.0)

The API will return a Pickup element in the OrderInformationReply response. The Pickup element contains the pickup confirmation number of the pickup request. The attribute confirmationNumber is optional.

```
<Pickup confirmationNumber ="123456789" />
```

3. OrderDetails (since v3.1.0).

If detailed attributes is set with true value in request then API will return OrderDetails element in OrderInformationReply response.

8. Services

The QuoteRequest and ShippingRequest messages have an optional attributed called "serviceId". The value of this attribute maps to one of several services offered by EShipper. A list of the services and corresponding values are shown below. If this attribute is set to a positive value other than 0 in the QuoteRequest, the EShipper application will return a quote for that service only. Similarly, if this attribute is set to a positive value other than 0 in the ShippingRequest, the application will try and ship the order using that service. If the service is not available or an invalid service id is supplied, no quote will be returned.

It is highly recommended to send a serviceId value with a ShippingRequest message. If no serviceId is sent with a ShippingRequest, EShipper will ship the shipment using a service of its discretion.

Service ID	Service Name	Shipment Country of Origin
1	FedEx Priority	CA
2	FedEx First Overnight	CA
3	FedEx Ground	CA

28	FedEx Standard Overnight	CA
29	FedEx 2nd Day	CA
30	FedEx Economy	CA
4	Purolator Express	CA
5	Purolator Express 9AM	CA
6	Purolator Express 1030	CA
7	Puroletter	CA
8	Puroletter 9AM	CA
9	Puroletter 1030	CA
10	PuroPak	CA
11	PuroPak 9AM	CA
12	PuroPak 1030	CA
13	Purolator Ground	CA
19	Purolator Ground 9AM	CA
20	Purolator Ground 1030	CA
100	DHL Express 9 AM	CA
101	DHL Express Worldwide	CA, US
102	DHL Express 10 30 AM	CA
103	DHL Express 12 PM	CA
109	DHL Express Envelope	US
600	UPS Express, Next Day Air	CA, US
601	UPS Expedited, Second Day Air	CA, US
602	UPS Worldwide Express	CA
603 604	UPS Worldwide Expedited UPS Standard (Ground)	CA CA
605	UPS Express Early A.M., Next Day Air Early A.M.	CA, US
606	UPS Three-Day Select	CA

607	UPS Saver	CA
608	UPS Ground	US
609	Next Day Saver	US
610	Worldwide Express Plus	US
611	Second Day Air A.M.	US
500	Canada Post Priority	CA
501	Canada Post Xpress Post	CA
502	Canada Post Expedited	CA
503	Canada Post Regular	CA
504	Canada Post Xpress Post USA	CA
505	Canada Post Xpress Post Int'l	CA
506	Canada Post Air Parcel Int'l	CA
507	Canada Post Surface Parcel Int'l	CA
508	Canada Post Expedited Parcel USA	CA
2700	Aramex Priority Letter Express	CA
2701	Aramex Priority Parcel Express	CA
2702	Aramex Economy Parcel Express	CA
2703	Aramex Economy Document Express	CA
3500	Global Mail Parcel Priority	US,CA
3501	Global Mail Parcel Standard	US,CA
3502	Global Mail Packet Plus Priority	US,CA
3503	Global Mail Packet Priority	US,CA
3504	Global Mail Packet Standard	US,CA
3505	Global Mail Business Priority	US,CA
3506	Global Mail Business Standard	US,CA
3507	Global Mail Parcel Direct Priority	US,CA

3508	Global Mail Parcel Direct Standard	US,CA
3602	Sameday Ground Service	CA
3606	Sameday LTL Service	CA
3607	H1 Deliver to Curbside	CA
3608	H2 Delivery to Room of Choice	CA
3609	H3 Delivery & Packaging Removal	CA
3610	H4 Delivery to Curbside	CA
3611	H5 Delivery to Room of Choice – 2 Man	CA
3612	H6 Delivery & Packaging Removal – 2 Man	CA
4000	Holland Motor Express	CA, US
4002	YRC	CA, US
4003	New Penn Motor Express`	US
4004	UPS`	US
4005	YRC`	CA, US
4006	Con-way Freight`	US
4007	R+L Carriers`	US
4008	FedEx Freight East`	CA, US
4009	FedEx National`	CA, US
4010	Estes Express Lines`	US
4011	ForwardAir`	US
4012	New England Motor Freight`	US
4013	New England Motor Freight	US
4014	Vision Express LTL`	US
4015	Central Freight Lines`	US
4016	SAIA Motor Freight`	US
4017	Roadrunner Transportation Services`	US
4018	Holland Motor Express`	US
4019	Fedex Freight Canada`	CA, US

4021	Dayton Freight Lines`	US
4022	UPS	US
4024	Midwest Motor Express`	US
4034	Central Transport`	US
4037	WARD Trucking`	US
4040	Dependable Highway Express`	US
4043	Polaris	US
4044	Clear Lane Freight`	US
4045	Frontline Freight`	US
4047	TST Overland Express	US
4048	Day & Ross	CA, US
4054	Fedex Freight Canada	CA
4055	Fedex Freight USA	US
4056	FedEx Freight National Canada	CA
4057	FedEx Freight National USA	US
4059	XPO Logistics	CA, US
4060	ABFS	CA, US
4200	Vitran Rail	CA
4201	Vitran Road	CA
4202	National Fastfreight Rail	CA
4203	Western Canada Rail	CA
4212	MO Rail	CA
4208	National Fastfreight Road	CA
4500	Canpar Ground	CA
4502	Canpar Select Letter	CA
4503	Canpar Select Pak	CA
4504	Canpar Select Parcel	CA
4505	Canpar Express Letter	CA

4506	Canpar Express Pak	CA
4507	Canpar Express Parcel	CA
4600	Kindersley Rail	CA
4602	Kindersley Expedited	CA
4603	Kindersley Express	CA
4800	Speedy	CA
4901	Envoi - Same Day Delivery	CA
5100	CPX	CA

PLEASE NOTE:

1. Purolator and Canada Post services are available for shipments originating in Canada only.
2. DHL International service is available for shipments originating in the United States only.
3. Federal Express and UPS services are available originating in both United States and Canada.
4. TST LTL service is available originating in Canada only.

The following additional services can be requested with the Quote and Shipping request.

1. COD and Insurance

All insurance amounts are to be requested at the package level (as opposed to order level). The use of “insuranceAmount” in the root element has been deprecated as of v2.0.9. For “Quote” and

“Shipping” requests, insurance amounts will have to be sent in the <Package> element in the attribute “insuranceAmount”. This amount has to be a float number.

For COD requests, the COD element will be sent which highlights the payment details such as payment type, and return address for COD (see examples in section 4 and 5). The actual amounts for COD are to be sent along with the package information in the <Package> element.

Note: The <CODReturnAddress> element needs to be sent only if the COD is to be shipped to a location different than the address specified in the <From> element, i.e. to a location other than the sender. Currently, this service (return COD to address other than sender) is available through FedEx USA only. In all other cases, COD will be returned to sender address.

Examples of the <Package> element for quotes and shipping requests containing insurance and COD amounts:-

Example 1

```
<Packages type="Package">

     <Package length="15" width="10" height="12"
     weight="12" codAmount=”100” insuranceAmount=”300”
     />

     <Package length="15" width="10" height="10"
     weight="14" codAmount=”120.54”/>

</Packages>
```

In this example, package1 has a COD amount of $100 and an insurance amount of $30.

Package2 has a COD amount of $120.54 and no insurance amount.

Example 2

```
<Packages type="Package">

     <Package length="15" width="10" height="12"
     weight="12"/>

     <Package length="15" width="10" height="10"
     weight="14" insuranceAmount=”250.0”/>

</Packages>
```

In this example, package1 has no COD and insurance amounts.

Package2 has no COD amount and an insurance amount of $250.

Example 3

```
<Packages type="Package">

     <Package length="15" width="10" height="12"
     weight="12"/>

     <Package length="15" width="10" height="10"
     weight="14"/>

</Packages>
```

In this example, both packages have no COD and insurance amounts.

2. Scheduled Ship Date

Quote and Shipping requests can include a “scheduledShipDate” field. This date can be the current date or a future date. The “scheduledShipDate” attribute can be sent in the <QuoteRequest> or <ShippingRequest> element. This attribute is optional.

The format for the date is yyyy-MM-dd (for example 2007-01-11 would imply January 11, 2007).

Example of scheduled date in the ShippingRequest element:-

```
<ShippingRequest serviceId=”100”
scheduledShipDate=”2007-02-14”>
```

3. Saturday Service Request

Saturday service can be requested by setting the “isSaturdayService” attribute in the Quote and Shipping requests. The “isSaturdayService” attribute can be sent in the <QuoteRequest> or <ShippingRequest> element. Valid values are “true” and “false”. This attribute is optional.

Example of Saturday service request in the ShippingRequest element:-

```
<ShippingRequest serviceId=”4” isSaturdayService=”true”>
```

4. Pickup (since v3.0.0)

Note: Pickup service is available only for UPS and Purolator.

Pickup service can be requested by setting the “Pickup” element in the Quote and Shipping requests. The “Pickup” element can be sent in the <QuoteRequest> or <ShippingRequest> element. This element is optional.

The format for the date is yyyy-MM-dd (for example 2007-01-11 would imply January 11, 2007).

The format for the time is HH:MM (for example 16:30 would imply 4:30 PM).

In case of UPS, phone number should contain at least 10 digits.

In case of Purolator, pickup location is required and format of phone number should be (###-###-####).

Valid values for location are:

1. Receiving
2. Garage
3. Lobby
4. Reception
5. Front Desk
6. Vault
7. Switchboard
8. Back Door
9. Desk
10. Between Doors
11. Kiosk
12. Office
13. Outside Door
14. Mailbox
15. Side Door
16. Service Counter
17. Security
18. Shipping
19. Front Door
20. Basement
21. Mail Room
22. Lab
23. Warehouse
24. Pharmacy
25. Pro Shop
26. Parts Department
27. Counter
28. Loading Dock
29. Gate House

Example of the <Pickup> element for quotes and shipping requests containing pickup:-

```
<Pickup contactName="Test Name" phoneNumber="888-888- 8888"
pickupDate="2009-08-03" pickupTime="16:30" closingTime="17:45"
location="Front Door" />
```

5. Signature Required Service (since v3.0.0)

Signature required service can be requested by setting the “signatureRequired” attribute in the Quote and Shipping requests. The “signatureRequired” attribute can be sent in the <QuoteRequest> or <ShippingRequest> element. This attribute is optional.

Valid values are “No” and “Yes”, if from country is CANADA.

Valid values are “No”, “Indirect”, “Direct” and “Adult”, if from country is US.

Example of Signature required service request in the ShippingRequest element:-

```
<ShippingRequest serviceId="4" signatureRequired="Yes">
```

Note: Only for UPS, Signature service not allowed with COD service.

2. *Hold for Pickup Required Service (since v3.0.0)*

Hold for Pickup Required service can be requested by setting the “holdForPickupRequired” attribute in the Quote and Shipping requests. The “holdForPickupRequired” attribute can be sent in the <QuoteRequest> or <ShippingRequest> element. Valid values are “true” and “false”. This attribute is optional.

Example of Hold for Pickup Required service request in the ShippingRequest element:-

```
<ShippingRequest serviceId="4" holdForPickupRequired="true">
```

6. Delivery Appointment Service (since v3.0.0)

Delivery Appointment service can be requested by setting the “deliveryAppointment” attribute in the Quote and Shipping requests if the package type is set to Pallet. The “deliveryAppointment” attribute can be sent in the <QuoteRequest> or <ShippingRequest> element. Valid values are “true” and “false”. This attribute is optional.

Example of Delivery Appointment service request in the ShippingRequest element:-

```
<ShippingRequest serviceId="4" deliveryAppointment="true">
```

7. Special Equipment Service (since v3.0.0)

Special Equipment service can be requested by setting the “specialEquipment” attribute in the Quote and Shipping requests if the package type is set to Pallet. The “specialEquipment” attribute can be sent in the <QuoteRequest> or <ShippingRequest> element. This attribute is optional.

Valid values are:

1) Container
2) Dry Box – 48'
3) Dry Box – 53'
4) Flat Bed – 48'
5) Flat Bed – 53'
6) Flat 4X – 48'
7) Reefer – 48'
8) Reefer – 53'
9) Straight Truck
10) Tractor
11) Trailer
12) Tri-Axle Van
13) Volvos

Example of Special Equipment service request in the ShippingRequest element:-

```
<ShippingRequest serviceId="4" specialEquipment="Container">
```

8. Inside Delivery Service (since v3.0.0)

Inside Delivery service can be requested by setting the “insideDelivery” attribute in the Quote and Shipping requests if the package type is set to Pallet. The “insideDelivery” attribute can be sent in the <QuoteRequest> or <ShippingRequest> element. Valid values are “true” and “false”. This attribute is optional.

Example of Inside Delivery service request in the ShippingRequest element:-

```
<ShippingRequest serviceId="4" insideDelivery="true">
```

9. Freight Class and Type (since v3.0.0)

All Freight Class and Type are to be requested at the package level if package type is set to Pallet. For “Quote” and “Shipping” requests, freight class and type will be sent in the <Package> elements as the attributes “freightClass” and type. These attributes are required if package type is set to Pallet.

Valid values for Freight Class are:

1) 50
2) 55
3) 60
4) 65
5) 70
6) 77
7) 77.5
8) 85
9) 92.5
10) 100
11) 110

12) 125
13) 150
14) 175
15) 200
16) 250
17) 300
18) 400

Valid values for Type are:

1) Pallet
2) Drum
3) Boxes
4) Rolls
5) Pipes/Tubes
6) Bales
7) Bags
8) Cylinder
9) Pails
10) Reels

Examples of the <Package> element for quotes and shipping requests containing freightClass and Type:-

```
<Packages type="Pallet">

           <Package length="15" width="10" height="12"
           weight="12" type=”Drum” freightClass=”70”/>

           <Package length="15" width="10" height="10"
           weight="14" type=”Boxes” freightClass=”77.5”/>
</Packages>
```

In this example, package1 has a type of “Drum” and freightClass of “70”.

Package2 has a type of “Boxes” and freightClass of “77.5”

10. Insurance Type (since v3.0.0)

Insurance Type service can be requested by setting the “insuranceType” attribute in the Quote and Shipping requests. The “insuranceType” attribute can be sent in the <QuoteRequest> or <ShippingRequest> element. This attribute is optional.

For envelope and pak, the only possible insurance type is 'Carrier'. In this case the insurance type will always be Carrier insurance and the maximum value will be $100.

For pallet, the only possible insurance type is ‘eShipper’. In this case the insurance type will always be eShipper insurance and the maximum value will be $100,000.

For package, the possible insurance types are 'eShipper' and 'Carrier'. In this case 'insuranceType' attribute is missing in the request then default customer insurance preference will be applied. The maximum allowed value will be $100,000.

Valid values are:

1. eShipper.
2. Carrier.

Example of Insurance Type service request in the ShippingRequest element:-

```
<ShippingRequest serviceId="4" insuranceType="Carrier">
```

11. Inside Pickup Service (since v3.0.4)

Inside Pickup service can be requested by setting the "insidePickup" attribute in the Quote and Shipping requests if the package type is set to Pallet. The "insidePickup" attribute can be sent in the <QuoteRequest> or <ShippingRequest> element. Valid values are "true" and "false". This attribute is optional.

Example of Inside Pickup service request in the ShippingRequest element:-

```
<ShippingRequest serviceId="4" insidePickup="true">
```

12. Residential Pickup Service (since v3.0.4)

Residential Pickup service can be requested by setting the "residential" attribute in the Quote and Shipping requests inside from element if the package type is set to Pallet. The "residential" attribute can be sent in the <Form> element. Valid values are "true" and "false". This attribute is optional.

Example of Residential Pickup service request in the From element:-

```
<From id="123" company="Test Company" address1="650 CIT Drive"
city="Livingston" state="ON" zip="L8E5X9" country="CA" phone="9052223333"
attention="Riz" email="riz@shaw.ca" residential="true" />
```

NOTE: A fee usually applies to Shipping Requests that require additional services.

9. 4D Boxing

4D Boxing API is used to get the appropriate Package Dimensions that could be used to get the optimized shipping cost based on the dimensions of the available Boxes.

4D Boxing Request consists of the following data:

a. Product details of all the Products that need to be shipped in one Request.

   This data comes under <ITEM> tag with attributes – Product Name, Length, Width, Height, Quantity and Weight.

b. Box details for all the Boxes that Customer uses for Shipping his Products.

   This data comes under <BOX> tag with attributes – Box Name, Length, Width, Height, Weight of the box and Maximum Weight a box can support.

4D Boxing Response will comprise of the following data:

a. Appropriate Box(es) with its Dimensions and Weight

   Corresponding to every Box, the Response API will list all the products that are chosen to be a part of this Box.

b. Custom Item(s) with their Dimensions and Weight

   These are the items which could not be accommodated with other products in a Box and hence need to be shipped individually. This data comes under <Custom Items> tag.

Sample Request/Response:

SAMPLE – 1

4D Boxing Request:

```
<?xml version="1.0" ?>

<EShipper xmlns="https://web.eshipper.com/rpc2" username="merchant
inc." password="9999" version="3.0.0">

        <BoxingRequest>

                <Items>

                    <Item name="SPRI Braided Xertube - Level 1 4" quantity="1"
length="10" width="10" height="10" weight="1"></Item>

                    <Item name="SPRI Braided Xertube - Level 2 4" quantity="1"
length="10" width="10" height="10" weight="1"></Item>

                </Items>

                <Boxes>

                    <Box name="Box1" quantity="1" length="20" width="20"
height="20" weight="1" max="10"></Box>

                    <Box name="Box2" quantity="1" length="10" width="10"
height="10" weight="1" max="5"></Box>
                </Boxes>

</BoxingRequest>
```

```
</EShipper>
```

4D Boxing Response:

```
<EShipper xmlns="https://web.eshipper.com/rpc2" version="3.0.0">

    <BoxingRequestReply>

        <BoxingRequestData result="Success" />

        <PackedBoxes>
            <Box name="Box1" length="20.0" width="20.0" height="20.0"
weight="3.0" >
                <Items>

                    <Item name="SPRI Braided Xertube - Level 1 4" length="10.0"
width="10.0" height="10.0" weight="1.0" quantity="1" />

                    <Item name="SPRI Braided Xertube - Level 2 4" length="10.0"
width="10.0" height="10.0" weight="1.0" quantity="1" />

                </Items>

            </Box>

        </PackedBoxes>

    </BoxingRequestReply>

</EShipper>
```

SAMPLE – 2

4D Boxing Request:

```
<?xml version="1.0" ?>

<EShipper xmlns="https://web.eshipper.com/rpc2" username="merchant
inc." password="9999" version="3.0.0">

<BoxingRequest>

  <Items>

    <Item name="SPRI Braided Xertube - Level 1 4" quantity="1" length="10"
  width="10" height="10" weight="1"></Item>

    <Item name="SPRI Braided Xertube - Level 2 4" quantity="1" length="10"
  width="10" height="10" weight="1"></Item>

  </Items>

  <Boxes>

    <Box name="Box1" quantity="1" length="15" width="15" height="15" weight="1"
  max="10"></Box>

    <Box name="Box2" quantity="1" length="10" width="10" height="10" weight="1"
  max="5"></Box>

  </Boxes>
```

```
</BoxingRequest>

</EShipper>
```

4D Boxing Response:

```
<EShipper xmlns="https://web.eshipper.com/rpc2" version="3.0.0">

    <BoxingRequestReply>

        <BoxingRequestData result="Success" />

        <PackedBoxes></PackedBoxes>

        <CustomItems>

            <Item name="SPRI Braided Xertube - Level 1 4" length="10.0"
width="10.0" height="10.0" weight="1.0" quantity="1" />

            <Item name="SPRI Braided Xertube - Level 2 4" length="10.0"
width="10.0" height="10.0" weight="1.0" quantity="1" />

        </CustomItems>

    </BoxingRequestReply>

</EShipper>
```

10.Invoice Request

In order to receive a invoice detail/summary from the EShipper API, the client can send a "InvoiceRequest" message to the API.The API will process the request and return a "**InvoiceRequestReply**" message.**To get invoice details set detailed="true" in request.To get invoice summary set detailed="false" in request.The invoice details/summary in the InvoiceRequestReply message are located between the <Invoice></Invoice> tag in base-64 encoded format. This string of characters needs to be decoded using a base-64 decryption algorithm, and the resulting bytes can then be written into a PDF file.** A sample request and response is shown below:-

Available Tags with Attributes in the InvoiceRequest

- InvoiceRequest [Required]
    - invoiceId [Required]
    - detailed [Optional]

Sample InvoiceRequest message

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

```
    <EShipper xmlns:es="http://www.eshipper.net/XMLSchema"
    username="upsscs" password="1234" version="3.1.0">


          <InvoiceRequest invoiceId="CWS0097055" detailed="true">


          </InvoiceRequest>


    </EShipper>
```

Sample InvoiceRequestReply response to InvoiceRequest message

```
<EShipper xmlns="http://www.eshipper.net/xml/XMLSchema"
version="3.1.0">

      <InvoiceRequestReply>

            <Invoice>[base-64 encoded String]</Invoice>

      </InvoiceRequestReply>

</Eshipper>
```

11. Order Search Request.

In order to receive orders information from the EShipper API, the client can send a "**OrderSearchRequest**" message to the API.The API will process the request and return a "**OrderSearchReply"** message.For a list of carrier mappings, please see Appendix D, section 16.1.

For a list of status mappings, please see Appendix B, section 10.1.

A sample request and response is shown below:

Available Tags with Attributes in the OrderSearchRequest

- OrderSearchRequest [Required]
- Order [Required]
    - fromDate [Required]
    - toDate [Required]
    - carrierId [Optional]
    - statusId [Optional]
    - startOrderId [Optional]
    - endOrderId [Optional]
    - carrierTrackingNumber [Optional]
    - referenceValue [Optional]

Sample OrderSearchRequest message

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

<EShipper xmlns="http://www.eshipper.net/XMLSchema" username="sumasoft"
password="1234" version="3.1.0">

      <OrderSearchRequest>

                  <Order fromDate="2010-03-01" toDate="2010-03-31"
carrierId="1" statusId="1" startOrderId="949984" endOrderId="949984"
carrierTrackingNumber="1" referenceValue="123"/>

      </OrderSearchRequest>

</Eshipper>
```

Sample OrderSearchReply response to OrderSearchRequest message

```
<EShipper xmlns="http://www.eshipper.net/xml/XMLSchema"
version="3.1.0">
```

```
<OrderSearchReply>

      <Order Id="949980">

            <References>

                  <Reference Code="" Name="null" />

            </References>

            <Service Id="4" Name="Purolator Air"/>

            <Carrier Id="2" Name="Purolator"/>

            <ShipmentDate Date="2010-03-17" />

            <ShipFrom Company="SumaSoft" Address1="My address"
            Address2="" City="Thornhill" Province="ON"
            PostalCode="L4J7Y9" Country="CA" />

            <ShipTo Company="SumaSoft2" Address1="My address"
            Address2="35" City="Burnaby" Province="BC"
            PostalCode="V3N4R3" Country="CA" />

            <Status StatusId="1" StatusName="READY FOR SHIPPING" />

      </Order>

</OrderSearchReply>

</Eshipper>
```

12. Invoice Payment Request.

In order to process Invoice payment request from the EShipper API, the client can send a “**InvoicePaymenthRequest**” message to the API.The API will process the request and return a “**InvoicePaymentReply**” message**.**

A sample request and response is shown below:-

Available Tags with Attributes in the InvoicePaymentRequest

- InvoicePaymentRequest [Required]
- CreditCard [Optional]

  - ExpiryMonth [Required]
  - ExpiryYear [Required]
  - Number [Required]
- Invoice
  - Id [Required]

Sample InvoicePaymentRequest message

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


<EShipper xmlns:es="http://www.eshipper.net/XMLSchema" username="demo"
password="1234" version="3.1.0">

<InvoicePaymentRequest>

<CreditCard ExpiryMonth=""  ExpiryYear=""  Number="" />


<Invoice Id="CWS0100024"/>

<Invoice Id="CWS0100123"/>


</InvoicePaymentRequest>


</EShipper>
```

Sample InvoicePaymentReply response to InvoicePaymentRequest message

```
<EShipper xmlns="http://www.eshipper.net/xml/XMLSchema"
version="3.1.0">

<InvoiceSearchReply>

<Invoice Id="CWS0101949" BalanceDue="$0.00" PaymentStatus="Paid"
AuthNumber="313624" Status="" />
```

```
<Invoice Id="CWS0102379" BalanceDue="$0.00" PaymentStatus="Paid"
AuthNumber="237632" Status=""/>

</InvoiceSearchReply>

</EShipper>
```

13. Pickup Request.

In order to schedule a pick up for an existing shipment from the EShipper API, the client can send a "PickupRequest" message to the API.. The API will process the request and return a "PickupReply" message.

A sample pickup request and response is shown below:-

Available Tags with Attributes in the PickupRequest

- Order [Required]
  - orderId [Required]
- Pickup [Required]
  - id [optional]
  - company [optional]
  - address1 [optional]
  - address2 [optional]
  - city [optional]
  - state [optional]
  - zip [optional]
  - country [optional]

contactName [Required]

phoneNumber [Required]

email [optional]

pickupDate [Required]

pickupReference [optional]

pickupTime [Required]

closingTime [Required]

location [Optional] For valid values refer section 8.4

instructions [optional]

Sample PickupRequest message

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

<EShipper xmlns="http://www.eshipper.net/XMLSchema"
   username="merchant inc." password="abcd" version="3.0.0">

   <PickupRequest>

         <Order orderId="383363"/>

         <Pickup id="123" company="Test Company" address1="650 CIT
   Drive" address2="Suite 103" city="Livingston" state="ON"
   zip="L8E5X9" country="CA" contactName="Test Name"
   phoneNumber="888-888-8888" email="riz@shaw.ca"

         pickupDate="2013-08-03" pickupReference="12345"
   pickupTime="16:30"

         closingTime="17:45" location="Front Door" instructions="my
   instructions"/>

   </PickupRequest>

 </EShipper>
```

Sample PickupReply response to PickupRequest message

```
<EShipper xmlns="http://www.eshipper.net/xml/XMLSchema"
         version="3.0.0">

   <PickupReply>

         <Order orderId="383363"
         pickupConfirmationNumber="2929602E9CP" />


         <Message>Pick up has been scheduled</Message>

   </PickupReply>

</EShipper>
```

14. Appendix A

Reading and Decoding the Labels from the ShippingReply

The labels in the ShippingReply message are located between the <Label></Labels> tag in base-64 encoded format. This string of characters needs to be decoded using a base-64 decryption algorithm, and the resulting bytes can then be written into a PDF file.

1. Java Example

In the Java programming language, this can be accomplished using the Base64 class found in the Apache org.apache.commons.codec.binary package.

Assuming that the String inside the labels tag has been read into a variable called 'encodedString', the following function can be called to create the pdf file:

```
public void generatePdf(String encodedString){

        try{

        FileOutputStream fos = new FileOutputStream(new File("C:\\temp\\mylabel.pdf"));

        byte[] bytes2 = Base64.decodeBase64(encodedString.getBytes());

        fos.write(bytes2);

        fos.close();

        }

        catch (Exception e) {

                e.printStackTrace();

        }

}
```

2. Other Languages

.NET

Look up the Convert class in System. It contains ToBase64String(), ToBase64CharArray(), FromBase64String(), and FromBase64CharArray() static methods.

C programming Language http://www.scripting.com/midas/

base64/source.html

15. Appendix B

10.1 Status Mappings

Status Id	Status
1	READY FOR SHIPPING
2	IN TRANSIT
3	DELIVERED
4	CANCELLED
5	EXCEPTION
7	CLOSED
9	READY FOR CHECKOUT (DHLeC)
10	READY TO PROCESS (DHLeC)

2. Invoice Search Request

In order to receive a invoice information from the EShipper API, the client can send a "InvoiceSearchRequest" message to the API.The API will process the request and return a "InvoiceSearchReply" message.For a list of status mappings, please see Appendix C, section 12.1.

A sample request and response is shown below:-

Available Tags with Attributes in the InvoiceSearchRequest

- InvoiceSearchRequest [Required]
  - fromDate [Required]
  - toDate [Required]
  - invoiceId [Optional]
  - woId [Optional]
  - statusId [Optional]

Sample InvoiceSearchRequest message

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

<EShipper xmlns:es="http://www.eshipper.net/XMLSchema"
username="FreightcomAPI" password="1234" version="3.1.0">

<InvoiceSearchRequest     fromDate="2010-10-28"
toDate="2010-10-28" invoiceId="CWS0132235" woId="1139752"
statusId="0">

</InvoiceSearchRequest>

</EShipper>
```

Sample InvoiceSearchReply response to InvoiceSearchRequest message

```
<EShipper xmlns="http://www.eshipper.net/xml/XMLSchema"
version="3.1.0">

<InvoiceSearchReply>

      <Invoice Id="CWS0132235" Date="2010-10-28 00:00:00.0" Age="1"
      No.ofWorkorders="1" Currency="CAD" Status="Unpaid" StatusId="0"
```

```
Charges="$14.98" Taxes="$1.71" Credits="$0.00"
Adjustments="$0.00" Total="$16.69" PaidAmount="$0.00"
BalanceDue="$16.69" Status2="">

      <WorkOrders>

            <WorkOrder oid="1139752" currencyCode="1" type="EDI"
            trackingNumber="329014353158" createdDate="2010-10-28
            11:43:23.0" shipDate="2010-10-28 00:00:00.0"
            cutomerName="FreightcomAPI" actualWeight="45.0"
            chargedWeight="0.0" status="INVOICED" packageId="3"
            packageTypeName="Package" packageSize="0"
            ediInvoiceDate="2010-09-17 00:00:00.0"
            ediInvoiceNumber="409808119">

                  <Carrier carrierName="Purolator"
                  serviceName="Purolator Express" serviceId="4"
                  />

                  <ShipFrom Company="TEST ONLY" Address1="My
                  address1" Address2="" City="THORNHILL"
                  Province="ON" Postal="L4J7Y9" Country="CA"
                  Residential ="N" Phone="8888888888"
                  Email="test@test.co" Attn="Mary"
                  Instructions="" />

                  <ShipTo Company="TEST ONLY" Address1="My
                  address1" Address2="" City="THORNHILL"
                  Province="ON" Postal="L4J7Y9" Country="CA"
                  Residential ="N" Phone="8888888888"
                  Email="test@test.co" Attn="Joe"
                  Instructions="***PLEASE DO NOT DISPATCH***TEST
                  ONLY***" />

                  <EshipperOrder id="1158508"/>

                  <OriginalCharges
                  totalAmount="14.836900352859498">

                        <Charge name="baseCharge"
                        amount="12.390000343322754"/>

                        <Charge name="fuelSurcharge"
                        amount="0.7400000095367432"/>

                        <Charge name="HST" amount="1.7069"/>

                  </OriginalCharges>

                  <ApplicableCharges
                  totalCharge="16.68690037727356"
                  totalCost="1.850000023841858">
```

```
                        <Charge name="Residential Delivery
                        Surcharge" code="ZS10"
                        totalValueCharge="1.85"
                        totalValue="1.85"/>

                        <Charge name="Fuel Surcharge"
                        code="SURCHARGE_FUEL"
                        totalValueCharge="0.74"
                        totalValue="0.0"/>

                        <Charge name="Freight"
                        code="SURCHARGE_FREIGHT"
                        totalValueCharge="12.39"
                        totalValue="0.0"/>

                        <Tax name="GST/HST" code="SURCHARGE_HST"
                        totalValueCharge="1.7069"
                        totalValue="0.0"/>

                  </ApplicableCharges>

            </WorkOrder>

      </WorkOrders>

   </Invoice>

</InvoiceSearchReply>

</EShipper>
```

16. Appendix C

11.1 Invoice Status Mappings

Status Id	**Status**
-1	ALL
0	Unpaid

1	Paid
2	Partial Paid
3	Released
4	Cancelled

17. Appendix D

15.1 Carrier Mappings

Carrier Id	Name
5	Canada Post
3	CANADA WORLDWIDE INC.
4	DHL
15	eShipper Trucking
1	Federal Express
2	Purolator
11	TST
6	UPS
18	e-Shipper Trucking
35	DHL eCommerce

18. Revision History

Date	Version	Modifications
Oct 12, 2006	2.0.1	Added Purolator Ground 9AM and 1030 services.
Oct 30, 2006	2.0.2	Added all DHL services.
Jan 11, 2007	2.0.3	COD and Insurance Handling for DHL Services (See section 8.1).
Jan 11, 2007	2.0.3	Scheduled Ship Date option (See section 8.2).
Jan 11, 2007	2.0.3	Saturday Service option Services (See section 8.3).
Jun 20, 2007	2.0.4	Returning tracking URL in ShippingReply
Dec 5, 2007	2.0.5	• FedEx USA service Ids added • UPS services added
Jan 20, 2007	2.0.6	Canada Post services added.
Jan 24, 2007	2.0.7	• Added FedEx Express Saver Service. • Updated API and schema to return label image (PNG) if available (see section 5.4).
February 12, 2007	2.0.8	• Added new API call – ShipmentCancelRequest (see section 6). • Added new API call – OrderInformationRequest (see section 7).
April 15, 2008	2.0.9	• Insurance amounts and COD amounts to be sent at package level (as opposed to order level). See section 8.1
March 01, 2009	2.1.0	• Added UPS services in USA • Added LTL services in USA and Canada • Removed DHL USA Domestic Services. • Changed test server URL
	3.0.0	Added Version Management (See section 3.1)
	3.0.0	• Returning currency code attribute with Quote element.(See section 4.1) • Returning Original Charge and Applicable Charge

		with OrderInformationReply (See Section 7.1) • Returning Pickup confirmation number with ShippingReply and OrderInformationReply (See Section 5.6/7.2)
	3.0.0	Added additional services:- • Schedule Pickup (See section 8.4), • Signature Required (See section 8.5), • Hold for Pickup Required (See section 8.6), • Delivery Appointment (See section 8.7), • Special Equipment (See section 8.8), • Inside delivery (See section 8.9), • Freight Class and Type (See section 8.10) • Insurance Type (See section 8.11)
	3.0.2	Updated Status
November 13, 2009	3.0.3	Updated API and Schema to return tailgate charge in QuoteRequest.
December 08, 2009	3.0.4	Added additional services: • Inside Pickup (See section 8.12) • Residential Pickup (See section 8.13)
March 15, 2010	3.1.0	Added new API call – InvoiceSearchRequest (see section 11).
March 26,2010	3.1.0	Added new API call – InvoiceRequest (see section 13).
March 30,2010	3.1.0	Added the additional information in response such as pickupfrom,shipto etc.(see section 7)
April 02,2010	3.1.0	Added new API call – OrderSearchRequest (see section 14).
April 07,2010	3.1.0	Added new API call – InvoicePaymentRequest (see section 15).
July 1, 2010	3.1.1	Added "DutiesTaxes' element to Customs Invoice
Aug 25,2010	3.1.2	Added accessorials and new fields for pallet option in shippingRequest and quoteRequest.
Aug 25 2010	3.1.2	Added NMFC Code for pallet option only in shipping request and quoteRequest.

Aug 27 2010	3.1.2	Added carrier e-Shipper Trucking and all its services.
September 21 2010	3.1.2	Added New e-Shipper Trucking service Roadrunner
October 25 2010	3.1.2	Changes in OrderInformationReply to add more shipping information
October 29 2010	3.1.3	Updated sample InvoiceSearchRequest and InvoiceSearchReply.
November 26 2010	3.1.3	Updated sample InvoiceSearchReply to add EshipperOrder tag
April 22, 2011	3.1.4	Updated Stackable attribute in QuoteRequest and ShippingRequest.
Sep 13, 2011	3.1.5	Added accessorials Sort and Segregate Charge for package type pallet.
Jan 11,2012	3.1.5	Added stackable attribute to sample Quote Request and sample Shipping Request.
Jul 11,2012	3.1.6	In-bond Manifest. See section 5.2 for more information.
Nov 06,2013	3.1.7	Added new pickup API call – PickupRequest (see section 17).
Aug 21, 2015	3.1.8	Added DHL eC specific API shipment changes.
Oct 19, 2015	3.1.9	Changed test server URL
June 23, 2020	4.0.0	Added New APIs for 4D Boxing functionality Updated Shipping Request API to fetch Packing Slip Updated Order Information request API Updated Shipment Cancellation request API Updated list of Carriers and Services