FREIGHTCOM

FREIGHTCOM API DOCUMENTATION V3.2

Revision Log

Date	Version	Modifications
July 01, 2010	3.1.0	Created Freightcom API document.
Aug 24, 2010 Aug24 2010	3.1.1 3.1.1	Updated to add ready time , origin and destination close time in shipping request for pallet option only. Updated NMFCCode for Pallet option only in Shipping Request and Quote Request.
Aug 30, 2010	3.1.2	Adding small package carrier and services.
October 01, 2010	3.1.3	Updated to add work order details in InvoiceSearchReply
October 29, 2010	3.1.4	Updated sample InvoiceSearchRequest and InvoiceSearchReply.
April 22, 2011	3.1.5	Updated Stackable attribute in `QuoteRequest and ShippingRequest.`
July 20 , 2011	3.1.6	Adding Purolator Freight and its services.
Sep 12, 2011	3.1.7	Adding SFW and its services.
Nov 28,2011	3.1.8	Added polaris and its services,added notes1 and note2 to invoice search Reply and added carrier error message to quoteReply
Feb 07,2013	3.1.9	Added Bill To address tags and SCAC code in Shipment Response.
Feb 14,2013	3.2	Replace Carrier Name by word “Freightcom” for all API responses from freightcom, and deleted Service table from document.
May 29,2013	3.2.1	Added podName and podDate attributes under status tag.
March 3,2016	3.2.2	Changed the credentials to make it generic, added the new logo
July 8, 2019	3.2.3	Added Quote Pickup Request and Response

1. Introduction

This document outlines the details of the API specifications for integrating with the Freightcom application. The Freightcom 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.

The protocol of communication between the client and the Freightcom application will be based on XML technology. Currently, the Freightcom will handle 8 types of requests from the client, (i) Quote Request, (ii) Ship Request, (iii) Order Information Request, (iv) Cancel Order Request, (v) Invoice Search Request, (vi) Invoice Request (vii) Order Search Request, (viii)Invoice Payment Request.All requests will be sent via the client in XML format, and the Freightcom 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 Freightcom administrator or your sales representative to ensure that you have been set-up on the application. A special password will then be provided to the customer, which will be sent in the request along with the customer name. The Freightcom application will use the customer name and password to authenticate the request before processing it.

All requests will be sent to the URL https://app.freightcom.com/rpc2 **.**

Customers can also use the test server to post test requests. The URL for the test server is **https://test.freightcom.com/rpc2**

3. XML Schema

The XML schema for the Freightcom API is shown below. This schema contains definitions for elements and attributes used in the 16 message types:- (i) QuoteRequest, (ii) QuoteReply, (iii) ShippingRequest, (iv) ShippingReply, (v) ShipmentCancelRequest, (vi) ShipmentCancelReply, (vii) OrderInformationRequest, and (viii) OrderInformationReply, (ix) InvoiceSearchRequest, (x) InvoiceSearchReply, (xi) InvoiceRequest, (xii) InvoiceRequestReply, (xiii) OrderSearchRequest, (xiv) OrderSearchReply, (xv) InvoicePaymentRequest, (xvi) InvoicePaymentReply.

3.1 Version Management

For all requests, version information will have to be sent in the <Freightcom> element.

Example of version in the Freightcom element:-

```
<Freightcom xmlns="http://www.freightcom.net/xml/XMLSchema"
username="username" password="XXXXX" version="3.1.0">
```

XSD file containing the schema is included in this package.

4. Getting a Quote

In order to receive a quote from the Freightcom 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]
  - insideDelivery [Optional]
  - scheduledShipDate [Optional]
  - insuranceType [Required if insurance amount is entered]
  - saturdayPickupRequired [Optional]
  - residentialDelivery [Optional]
  - tailgateDelivery [Optional]
  - singleShipment [Optional]
  - notifyRecipient [Optional]
  - crossBorderFee [Optional]
  - residentialPickup [Optional]
  - tailgatePickup [Optional]
  - excessLength [Optional]
  - limitedAccess [Optional]
  - customsIn-bondFreight [Optional]
  - militaryBaseDelivery [Optional]
  - exhibitionConventionSite [Optional]
  - pierCharge [Optional]
  - homelandSecurity [Optional]
  - stackable [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]
- 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 [Optional][Required if type=”Package” or “Pallet”]
  - type [Required]
- Package [Optional][Required if type=”Package” or “Pallet”]
  - length [Required]
  - width [Required]
  - height [Required]
  - weight [Required]
  - 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”]

Sample QuoteRequest message

```
<?xml version="1.0" encoding="UTF-8"?>
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" username="username"
password="XXXX" version="3.1.0">
        <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" package information is not required, otherwise it must be
            included -->
            <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="XXXX56"
                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>
        </QuoteRequest>
</Freightcom>
```

Sample QuoteReply response to QuoteRequest message

```
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" version="3.1.0">
        <QuoteReply>
            <Quote carrierId="20" carrierName="Freightcom" serviceId="2029"
            serviceName="Central Transport" modeTransport="A" transitDays="1"
            baseCharge="177.0" fuelSurcharge="0.0" totalCharge="177.0"
            currency="CAD">
            </quote>

            <Quote carrierId="21" carrierName="Freightcom" serviceId="2107"
            serviceName="Estes" modeTransport="G" transitDays="1"
            baseCharge="28.650000000000002" fuelSurcharge="0.0" totalCharge="28.65"
            currency="CAD">
            </quote>

            <Quote carrierId="19" carrierName="Freightcom" serviceId="1911"
            serviceName="USF Holland" modeTransport="null" transitDays="0"
            baseCharge="46.27000045776367" fuelSurcharge="6.25" totalCharge="52.52"
            currency="CAD">
            </quote>

            <CarrierErrorMessage size="1" errorMessage0="Polaris:Military Base
            Delivery,Saturday Pickup,Construction Site,BORDER FEE,Homeland
            Security,Limited Access,Saturday Delivery,Sort and Segregate Charge,Pier
            Charge">
            </CarrierErrorMessage>

        </QuoteReply>
<Freightcom>
```

5. Shipping an Order

In order to ship an order using the Freightcom 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]
  - insideDelivery [Optional]
  - scheduledShipDate [Optional]
  - insuranceType [Required if insurance amount is entered]
  - saturdayPickupRequired [Optional]
  - residentialDelivery [Optional]
  - tailgateDelivery [Optional]
  - singleShipment [Optional]
  - notifyRecipient [Optional]
  - crossBorderFee [Optional]
  - residentialPickup [Optional]
  - tailgatePickup [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]
- 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 [Required]
  - instructions [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 [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 [Optional][Required if type=”Package” or “Pallet”]
  - type [Required]
- Package [Optional][Required if type=”Package” or “Pallet”]
  - length [Required]
  - width [Required]
  - height [Required]
  - weight [Required]
  - 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”]

- CustomsInvoice [Optional] [Required if International shipment.]
- 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]

Sample ShippingRequest message

```
<?xml version="1.0" encoding="UTF-8"?>
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" username="username"
password="XXXX" version="3.1.0">
        <ShippingRequest serviceId="2029" stackable="true" >
                <From id="123" company="Test Company" address1="650 CIT Drive"
                city="Livingston" state="ON" zip="L8E5X9" country="CA"
                phone="9052223333" attention="Test" email="test@test.com"
                residential="true"/>

                <To company="Test Company" address1="650 CIT Drive" city="Livingston"
                state="BC" zip="V3N4R3" country="CA" phone="4162223333"
                attention="TestTo" email="test@test.com"/>

                <!-- 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" package information is not required, otherwise it must be
                included -->
                <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" nmfcCode="XXXX56"
                        insuranceAmount="0.0" codAmount="0.0" description="desc."/>
                        <Package length="15" width="10" height="10" weight="14"
                        type="Pallet" freightClass="70" insuranceAmount="0.0"
                        codAmount="0.0" description="desc."/>
                </Packages>

                <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="XXXX56" />
                <Reference name="" code="" />
                <Reference name="" code="" />

                <!--CustomsInvoice is optional, required for international shipment-->
                <CustomsInvoice>
                        <BillTo company="ABC Towing" name="Alfred" address1="444
                        Highway 401" city="Toronto" state="ON" zip="A1B 2C3"
                        country="CA" />

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

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

        </ShippingRequest>
</Freightcom>
```

Sample ShippingReply response to ShippingRequest message

```
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" version="3.1.0">
        <ShippingReply>
            <Order id="181004" />
            <Carrier carrierName="Freightcom" serviceName="Central Transport"
            SCAC="RLCA" />
            <Reference code="XXXX567" name="TestReference" />
            <Package trackingNumber="052800410000484" />
            <Package trackingNumber="052800410000491" />

            <Pickup confirmationNumber ="XXXX56789" />

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

            <!--There will be only one quote element in the shipping reply-->
            <Quote carrierId="20" carrierName="Freightcom" serviceId="2029"
            serviceName="Central Transport" modeTransport="null" transitDays="0"
            baseCharge="30.739999771118164" fuelSurcharge="0.0" totalCharge="31.82"
            currency="CAD">
                <Surcharge id="null" name="Other" amount="1.0800000429153442"/>
```

```
				</quote>
				<!--Bill to address will be carrier (may be service) specific in the shipping reply-->
				<BillingAddress CompanyName="Freightcom Inc. FCPC (10)" Address1="7699 Bath Road" Address2="" City="Mississauga" ProvinceCode="ON" CountryCode="CA" zip="L4T3T1" PhoneNo="">
				</BillingAddress>
		</ShippingReply>
</Freightcom>
```

As seen in the request above, the serviceId was set to “2029” which implies that a “Central Transport” 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.

5.1 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.

5.2 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.

5.3 Label Images

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>
```

5.4 Quote

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

6. Shipment Cancel Request

In order to cancel an order that was previously created using the Freightcom API, the client can send a "ShipmentCancelRequest" message to the API. The system will attempt to cancel the order based on the 'orderId' 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.

<u>Sample ShipmentCancelRequest message</u>

```
<?xml version="1.0" encoding="UTF-8"?>
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" username="username"
password="XXXX" version="3.1.0">
        <ShipmentCancelRequest>
                <Order orderId="383363"/>
        </ShipmentCancelRequest>
</Freightcom>
```

<u>Sample ShipmentCancelReply response to ShipmentCancelRequest message</u>

```
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" version="3.1.0">
        <ShipmentCancelReply>
                <Order orderId="383363" message="Order has been cancelled!" />
                <Status statusId="4" />
        </ShipmentCancelReply>
</Freightcom>
```

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.

<u>Sample OrderInformationRequest</u>

```
<?xml version="1.0" encoding="UTF-8"?>
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" username="username"
password="XXXX" version="3.1.0">
        <OrderInformationRequest>
                <Order orderId="232212" detailed="true"/>
        </OrderInformationRequest>
</Freightcom>
```

Sample OrderInformationReply response to OrderInformationRequest message

```
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" version="3.1.0">
<OrderInformationReply>
<Order id="949986" />
<Carrier carrierName="Freightcom" serviceName="Central Transport" />
<Reference code="" name="" />
<Package trackingNumber="0-1" />
<TrackingURL/>
<Pickup />
<Labels>[base-64 encoded String]</Labels>

<!-- If status is In Transit then only podDate attribute will be sent and if status is delivered the
podName and podDate both will be sent. -->

<Status statusId="1"
statusName="DELIVERED" podName="TEST" podDate="2013-05-10 00:00"
/>
<Quote carrierId="20" carrierName="Freightcom" serviceId="2029" serviceName="Central
Transport" 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" Address1="My address" Address2="pune"
        Phone="8888888888" Email="test@test.co" Attn="mani" Instructions="Inst 1"/>
        <ShipTo Company="SumaSoft2" Address1="My address" Address2="35"
        Phone="8888888888" Email="test@test.co" Attn="Mani" Instructions="Inst 2"/>
        <Packaging Type="Packaging">
                <Package Length="3" Width="2" Height="1" Weight="4"
                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>
</Freightcom>
```

7.1 OriginalCharge and ApplicableCharge

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.

7.2 OrderDetails

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 Freightcom. 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 Freightcom 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, Freightcom will ship the shipment using a service of its discretion.

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

8.1 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.

8.2 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:-

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

8.3 Freight Class and Type

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”

8.4 Insurance Type

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. But Required if insurance amount is entered.

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

Valid values are:

1. freightcom

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

```
<ShipipngRequest serviceId="2029" insuranceType="freightcom">
```

9. 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.

9.1 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();
        }
}
```

9.2 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

10. Appendix B

10.1 Status Mappings

Status Id	Status
1	READY FOR SHIPPING
2	IN TRANSIT
3	DELIVERED

4	CANCELLED
5	EXCEPTION
7	CLOSED

11. Invoice Search Request

In order to receive a invoice information from the Freightcom 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"?>
<Freightcom xmlns="http://www.freightcom.net/xml/XMLSchema" username="FCApi"
password="XXXX" version="3.1.0">
<InvoiceSearchRequest fromDate="2010-10-10" toDate="2010-10-26"
invoiceId="CWS0000008" woId="172" statusId="0">
</InvoiceSearchRequest>
</Freightcom>
```

Sample InvoiceSearchReply response to InvoiceSearchRequest message

```
<Freightcom xmlns="http://www.freightcom.net/xml/XMLSchema" version="3.1.0">
<InvoiceSearchReply>
        <Invoice Id="CWS0000008" Date="2010-10-22 00:00:00.0" Age="7"
        No.ofWorkorders="1" Currency="USD" Status="Unpaid" StatusId="0"
        Charges="$96.38" Taxes="$0.00" Credits="$0.00" Adjustments="$0.01" Total="$96.38"
        PaidAmount="$0.00" BalanceDue="$96.38" Status2="">
        <WorkOrders>
                <WorkOrder oid="172" notes1="" notes2="" currencyCode="2" type="EDI"
                trackingNumber="401012662" createdDate="2010-10-22 08:55:20.0"
                shipDate="2010-09-01 00:00:00.0" cutomerName="FC USA"
                actualWeight="270.0" chargedWeight="270.0" status="INVOICED"
                packageId="4" packageTypeName="Pallet" packageSize="0"
                ediInvoiceDate="2010-09-15 00:00:00.0" ediInvoiceNumber="68308">
                        <Carrier carrierName="Freightcom
                        " serviceName="Conway" serviceId="2004" />
                        <ShipFrom Company="A1Equipment" Address1="My address1"
                        Address2="" City="MIAMI" Province="FL" Postal="33136"
```

```
            Country="US" Residential ="N" Phone="8888888888"
            Email="test@test.co" Attn="Ivette Padron" Instructions="" />
            <ShipTo Company="Pierogies/forgotton taste" Address1="My address1"
            Address2="Suite 406" City="WEXFORD" Province="PA"
            Postal="15090" Country="US" Residential ="N" Phone="8888888888"
            Email="test@test.co" Attn="Marek Kudla" Instructions="Please contact
            Mr. Kudla one hour prior to delivery." />
            <Bol id="20989927">
            </Bol>
            <OriginalCharges totalAmount="96.37000274658203">
                <Charge name="baseCharge" amount="86.94000244140625"/>
                <Charge name="fuelSurcharge"
                amount="9.430000305175781"/>
            </OriginalCharges>
            <ApplicableCharges totalCharge="96.37999725341797"
            totalCost="96.37999725341797">
                <Charge name="FUEL SURCHARGE" code="FSC"
                totalValueCharge="9.43"  totalValue="9.43"/>
                <Charge name="Freight Charge" code="FRT"
                totalValueCharge="86.95" totalValue="86.95"/>
            </ApplicableCharges>
        </WorkOrder>
    </WorkOrders>
</Invoice>
</InvoiceSearchReply>
</Freightcom>
```

12. Appendix C

12.1 Status Mappings

Status Id	**Status**
-1	ALL
0	Unpaid
1	Paid
2	Partial Paid
3	Released
4	Cancelled

13. Invoice Request

In order to receive a invoice detail/summary from the Freightcom 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"?>

        <Freightcom xmlns="http://www.freightcom.net/XMLSchema" username="username"
password="XXXX" version="3.1.0">

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

                </InvoiceRequest>

        </Freightcom>
```

Sample InvoiceRequestReply response to InvoiceRequest message

```
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" version="3.1.0">
        <InvoiceRequestReply>
                <Invoice>[base-64 encoded String]</Invoice>
        </InvoiceRequestReply>
</Freightcom>
```

14. Order Search Request

In order to receive orders information from the Freightcom 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 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"?>
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" username="username"
password="XXXX" 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>
</Freightcom>
```

Sample OrderSearchReply response to OrderSearchRequest message

```
<Freightcom xmlns="http://www.freightcom.net/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>
</Freightcom>
```

15. Invoice Payment Request

In order to process Invoice payment request from the Freightcom 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"?>

<Freightcom xmlns="http://www.freightcom.net/XMLSchema" username="username"
password="XXXX" version="3.1.0">
<InvoicePaymentRequest>
<CreditCard ExpiryMonth="" ExpiryYear="" Number="" />

<Invoice Id="CWS0100024"/>
<Invoice Id="CWS0100123"/>

</InvoicePaymentRequest>

</Freightcom>
```

Sample InvoicePaymentReply response to InvoicePaymentRequest message

```
<Freightcom xmlns="http://www.freightcom.net/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>
</Freightcom>
```

16. Quote Pickup Request

In order to process quote pickup request from the Freightcom API, the client can send a “**PickupRequest**” message to the API. The API will process the request and return a “**PickupResponse”** message**.**

A sample request and response is shown below:-

Available Tags with Attributes in the **PickupRequest**

- PickupRequest [Required]
  - orderId [Required]

- PickupAddress [Required]
  - address1 [Required]
  - address2 [Optional]
  - city [Required]
  - province [Required]
  - country [Required]
  - zip [Required]
  - packageLocation [Required]
- PickupInfo [Required]
  - pickupDate [Required]
  - pickupTime [Required]
  - closeTime [Required]

  - weight [Required]
  - numPackages [Required]
  - specialInstructions [Required]
- ContactInfo [Required]
  - name [Required]
  - phone [Required]

```
<?xml version="1.0" encoding="UTF-8"?>
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" username="username"
password="XXXX" version="3.1.0">
<PickupRequest orderId=”827714”>
<PickupAddress address1=”Verendrye Ave 222, 777” address2=” ” city=”Burnaby” province=”BC” country=”CA”
zip=”V3J1A2” packageLocation=”BackDoor” />

<PickupInfo pickupDate=”2018-02-03” pickupTime=”13:40” closeTime=”14:40” weight=”7.00”
numPackages=”1” specialInstructions=” ” />
<ContactInfo name=”Valentina Staging EU” phone=”1914961921” />

</PickupRequest >

</Freightcom>
```

Sample PickupResponse to PickupRequest message

```
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" version="3.1.0">
<PickupResponse>
<ConfirmationNum>2929602E9CP</ConfirmationNum>
</PickupResponse >
</Freightcom>
```

17. Cancel Pickup Request

In order to process cancel pickup request from the Freightcom API, the client can send a “**CancelPickupRequest**” message to the API. The API will process the request and return a “**CancelPickupResponse”** message**.**

A sample request and response is shown below:-

Available Tags with Attributes in the **CancelPickupRequest**

- CancelPickupRequest [Required]
  - orderId [Required]

```
<?xml version="1.0" encoding="UTF-8"?>
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" username="username"
password="XXXX" version="3.1.0">
<CancelPickupRequest orderId=”827714” >
</CancelPickupRequest >
</Freightcom>
```

Sample CancelPickupResponse to CancelPickupRequest message

```
<Freightcom xmlns="http://www.freightcom.net/XMLSchema" version="3.1.0">
<CancelPickupResponse>success</CancelPickupResponse >
</Freightcom>
```