Web Service Technical Documentation

Rate and Shipment Requests

Modified November 21, 2021

REQUEST "TYPES"

Type 1 (Web Service Request 1) – returns all carriers; rate shopping

Type 2 (Web Service Request 1) – returns cost values for a rated shipment and adds it to ARTC accrual file

Type 3 (Web Service Request 1) – returns one carrier (cheapest preferred)

Type 4 (Web Service Request 2) – returns freight amount by line item.

Type 5 (Web Service Request 3) – allows user to delate a shipment from ARTC accrual file.

Type 6 (Web Service Request 4) – returns a carrier pro number.

Web Service Request 1 (Types 1=Rate Shop; 2=Rate Shipment; 3=Preferred Carrier)

This is a RESTful web service. The XML-formatted request must be sent via an HTTP POST request to:
https://webservices.artraffic.com/upload/webservice1.php
The POST string must be named *datain*'.

XML Data Format:

The XML data must be encoded using UTF-8 as shown below. Red elements are required.

```
<?xml version="1.0" encoding="UTF-8"?>
<request> *The entire request must be enclosed in request tags
 <user>Assigned User Name</user> *3 alphanumeric characters
 <password>Assigned Password</password> *6 alphanumeric characters
    <type>1, 2 or 3</type> *1 numeric character 1= Rate Shop 2=Shipment Rating 3=Preferred Carrier
<carrier>Carrier Code</carrier> *4 alphanumeric characters REQUIRED WITH TYPE 2
<mode>Mode</mode> *1 alphanumeric character A=Air L=LTL P=Parcel T=TL For filtering TYPE 1
<equipmentcode>Equipment Code</equipmentcode> *2 alphanumeric characters
 <days>Days</days> *2 alphanumeric character 01 02 03 etc For filtering TYPE 1
 <direction>"I" for inbound; "O" for outbound</direction> *1 alphanumeric character
 <warehouse>Warehouse Code</warehouse> *up to 6 alphanumeric characters
 <terms>"A" for Prepay & Add; "C" for Collect; "P" for Prepay</terms> *1 alphanumeric character
 <shipdate>Shipment date (YYYYMMDD)</shipdate> *8 numeric characters
 <shipmentnum>Shipment #; Must be unique for each shipment</shipmentnum> *up to 17 alphanumeric characters
 <pronum>Pro #</pronum> *up to 15 alphanumeric characters
 <multistop>"Y" for Yes; "N" for No</multistop> *1 alphanumeric character
 <trucknum>Truck #</trucknum> *up to 10 alphanumeric characters
 <trailnernum>Trailer #</trailernum> *up to 10 alphanumeric characters
 <stop> *At least one stop per request
      <zip>Shipment Zip</zip> *up to 6 alphanumeric characters
      <city>Shipment City</city> *up to 25 alphanumeric characters
       <state>Shipment State</state> *2 alphanumeric characters
       <county>Shipment County</county> *2 alphanumeric characters
       <country>Shipment Country</country> *3 alphanumeric characters
       <accessorials>Accessorial Codes</accessorials> *up to 10 alphanumeric characters
```

These elements may only be specified once per request

<customernum>Customer #</customernum> *up to 10 alphanumeric characters
<customer>Customer Name</customer> *up to 30 alphanumeric characters
<customeraddress>Customer Address</customeraddress> *up to 30 alphanumeric characters
<subbl>Sub B/L</subbl> *up to 15 alphanumeric characters
<stopsequence>Stop Sequence</stopsequence> *up to 2 alphanumeric characters (1-99)
<detail> *At least one detail per stop
<duedate>Due Date (YYYYMMDD)</duedate> *8 numeric characters
<linenum>Line #</linenum> *up to 5 numeric characters
<class>NMFC Class</class> *up to 4 alphanumeric characters, *see below(**)*
<weight>Weight (lbs.) </weight> *up to 8 alphanumeric characters, *see below(***)*
<cube>Cube</cube> *up to 8 alphanumeric characters, *see below(***)*
<carton>Carton</carton> *up to 10 alphanumeric characters, *see below(****)*
<pallets>Pallets</pallets> *up to 8 alphanumeric characters, *see below(***)*
<points>Points</points> *up to 8 alphanumeric characters, *see below(***)*
<pieces>Pieces</pieces> *up to 10 alphanumeric characters, *see below(****)*
<cases>Cases</cases> *up to 8 alphanumeric characters, *see below(***)*
<skupartnum>SKU Part #</skupartnum> *up to 15 alphanumeric characters, *see below(Spot Quote procedure)*
<skupartvalue>SKU Part Value</skupartvalue> *up to 8 numeric characters (0-99,999,999) no decimal positions allowed
<ordernum>Order #</ordernum> *up to 20 alphanumeric characters
<generalledger>General Ledger</generalledger> *up to 15 alphanumeric characters
<length>Length</length> *up to 3 numeric characters (0-999) no decimal positions allowed
<width>Width</width> *up to 3 numeric characters (0-999) no decimal positions allowed
<height>Height</height> *up to 3 numeric characters (0-999) no decimal positions allowed
</detail>
</stop>
</request>

**Acceptable NMFC Classes: 50; 55; 60; 65; 70; 77.5; 85; 92.5; 100; 110, 125; 150; 175; 200; 250; 300; 400; 500

***Weight, Cube, Pallets, Points, Cases: Must be between 1 and 99999.99. Up to 2 decimal positions are allowed. There must be no thousands separator.

****Carton, Pieces: Must be between 1 and 9999999.99. Up to 2 decimal positions are allowed. There must be no thousands separator.

SPOT QUOTE PROCEDURE (SHIP REQUEST 2 ONLY):

If this is a spot quote, the element "skupartnum" must be formatted as this: <skupartnum>ZZZZZZ99999.99</skupartnum>

It must start with ZZZZZZ six capital letter z's and then the freight dollar amount of the shipment right adjusted and zero filled.

Example 1: a $1250.50 shipment would be loaded as: <skupartnum>ZZZZZZ01250.50</skupartnum>

Example 2: a $625.00 shipment would be loaded as: <skupartnum>ZZZZZZ00625.00</skupartnum>

Output:

All output data is formatted in XML and enclosed in an element named 'response'. There are 2 elements enclosed in 'response': 'code' and 'data'. 'Code' contains a response code that describes the status of the request. The response codes are:

Response Code	HTTP Status Code	Description
0	400 (Bad Request)	Unknown Error
1	200 (OK)	Success
2	403 (Forbidden)	HTTPS Required
3	401 (Unauthorized)	Authentication Required
4	401 (Unauthorized)	Authentication Failed
5	406 (Unacceptable Request)	Invalid Shipment Data String Format

'Data' contains the output of the web service request. If there is an error with the request it contains a description of the error. If there is an error with the data in the request, it returns an error code. See *errormsg.txt* for a listing of error codes.

If request 1 or 3 are successful and the data is valid, the format of the output string ('data') is: mode|scac|carrier|freight amount|adjusted amount (if applicable)|days|direct *or* interline|routing preference (if any)$

The end of a record (carrier) is indicated by: $. If there is/are more carrier(s), their data will follow in the same string.

If request 2 is successful and the data is valid, the format of the output string ('data') is:
freight amount|adjusted amount (if applicable)$
Please note with a request 2, decimals will be returned in the amount field(s).

SAMPLE WEBSERVICE 1, TYPE 1 REQUEST AND RESPONSE

SAMPLE TYPE 1 REQUEST

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
  <user>***</user>
  <password>******</password>
  <type>1</type>
  <direction>O</direction>
  <multistop>N</multistop>
  <warehouse>DL</warehouse>
  <shipdate>20211118</shipdate>
  <terms>P</terms>
  <multistop>N</multistop>
  <shipmentnum>A054144700</shipmentnum>
   <stop>
    <zip>10004</zip>
    <state>NY</state>
    <accessorials></accessorials>
    <customernum>100850</customernum>
    <stopsequence>1</stopsequence>
    <detail>
      <duedate>20211118</duedate>
      <carton>1</carton>
      <class>50</class>
      <weight>1000</weight>
      <Length>15</Length>
      <Width>12</Width>
      <Height>5</Height>
      <generalledger>21110-000-10-00</generalledger>
    </detail>
  </stop>
</request>
```

SAMPLE TYPE 1 RESPONSE

```
<response>
<code>1</code>
<data>L|FXFE|Fedex Freight Priority |0026290|0000000|04|D| |$L|FXNL|FedEx Economy |0027835|0000000|07|D|
|$L|ODFL|OLD DOMINION FREIGHT LINE|0032684|0000000|03|D| |$</data>
</response>
```

Web Service Request 2: (Type 4=Line Item/Sku and Order Cost allocation)

This is a RESTful web service. The XML-formatted request must be sent via an HTTP POST request to:
https://webservices.artraffic.com/upload/webservice2.php
The POST string must be named *datain*'.

XML Data Format:

The XML data must be encoded using UTF-8 as shown below. Red elements are required.

<?xml version="1.0" encoding="UTF-8"?>
<request> *The entire request must be enclosed in *request* tags
<user>Assigned User Name</user> *3 alphanumeric characters
<password>Assigned Password</password> *6 alphanumeric characters
<type>4</type> *1 numeric character 4= Line Item Freight Cost Allocation
<shipmentnum>Shipment #; Must be unique for each shipment</shipmentnum> *up to 17 alphanumeric characters
</request>

Output:

All output data is formatted in XML and enclosed in an element named 'response'. There are 2 elements enclosed in 'response': 'code' and 'data'. 'Code' contains a response code that describes the status of the request. The response codes are:

Response Code	HTTP Status Code	Description
0	400 (Bad Request)	Unknown Error
1	200 (OK)	Success
2	403 (Forbidden)	HTTPS Required
3	401 (Unauthorized)	Authentication Required
4	401 (Unauthorized)	Authentication Failed
5	406 (Unacceptable Request)	Invalid Shipment Data String Format

'Data' contains the output of the web service request. If there is an error with the request it contains a description of the error. If there is an error with the data in the request, it returns an error code. See *errormsg.txt* for a listing of error codes.

If request Type 4 is successful and the data is valid, the format of the output string ('data') is:
skupartnum|ordernum|carrier|freight amount|adjusted amount (if applicable)$

Note: Up to a maximum of 25 entries (skupartnum|ordernum) will be returned per call.

The end of a record (skupartnum ordernum) is indicated by: $. If there are more skupartnum ordernum(s), their data will follow in the same string.

Web Service Request 3: (Type 5=Delete Shipment from ARTC Accrual File)

This is a RESTful web service. The XML-formatted request must be sent via an HTTP POST request to:
https://webservices.artraffic.com/upload/webservice3.php
The POST string must be named *datain*'.

XML Data Format:
The XML data must be encoded using UTF-8 as shown below. Red elements are required.

```
<?xml version="1.0" encoding="UTF-8"?>
<request> *The entire request must be enclosed in request tags
  <user>Assigned User Name</user> *3 alphanumeric characters
  <password>Assigned Password</password> *6 alphanumeric characters
    <type>5</type> *1 numeric character 5= Shipment Delete (From BLRTEP)
  <shipmentnum>Shipment #; Must be unique for each shipment</shipmentnum> *up to 17 alphanumeric characters
  </request>
```

Output:
All output data is formatted in XML and enclosed in an element named 'response'. There are 2 elements enclosed in 'response': 'code' and 'data'. 'Code' contains a response code that describes the status of the request. The response codes are:

Response Code	HTTP Status Code	Description
0	400 (Bad Request)	Unknown Error
1	200 (OK)	Success
2	403 (Forbidden)	HTTPS Required
3	401 (Unauthorized)	Authentication Required
4	401 (Unauthorized)	Authentication Failed
5	406 (Unacceptable Request)	Invalid Shipment Data String Format

'Data' contains the output of the web service request. If there is an error with the request it contains a description of the error. If there is an error with the data in the request, it returns an error code. See *errormsg.txt* for a listing of error codes.

If this request Type 5 is successful and the data is valid, this the response:

```
<response>
<code>1</code>
<data>Shipment Deleted</data>
</response>
```

If request 5 is successful and the shipment is not found, this the response:

```
<response>
<code>1</code>
<data>Shipment Not Found</data>
</response>
```

Web Service Request 4: (Type 6=Carrier ProNumber)

This is a RESTful web service. The XML-formatted request must be sent via an HTTP POST request to:
https://webservices.artraffic.com/upload/webservice4.php
The POST string must be named datain'.

XML Data Format:

The XML data must be encoded using UTF-8 as shown below.

<?xml version="1.0" encoding="UTF-8"?>
<request> *The entire request must be enclosed in request tags
<user>Assigned User Name</user> *3 alphanumeric characters
<password>Assigned Password</password> *6 alphanumeric characters

<type>6</type> *1 numeric character 6=Pronum Request
<carrier>Carrier Code</carrier> up to 4 characters
<warehouse>Warehouse Code</warehouser> up to 6 characters
</request>

Output:
All output data is formatted in XML and enclosed in an element named 'response'. There are 2 elements enclosed in 'response': 'code' and 'data'. 'Code' contains a response code that describes the status of the request. The response codes are:

Response Code	HTTP Status Code	Description
0	400 (Bad Request)	Unknown Error
1	200 (OK)	Success
2	403 (Forbidden)	HTTPS Required
3	401 (Unauthorized)	Authentication Required
4	401 (Unauthorized)	Authentication Failed
5	406 (Unacceptable Request)	Invalid Shipment Data String Format

'Data' contains the output of the web service request.
If there is an error with the data in the request, it returns an error code. See errormsg.txt for a listing of error codes.

If this request Type 6 is successful and the data is valid, this the response:

```
<response>
<code>1</code>
<data>XXXXXXXXXXXXXXX</data> where XXXXXXXXXXXXXXX is the pro number
</response>
```