

FLIGHT ROUTE ALERTING AND BRIEFING

API – Version 1.83

Table of Contents

1	Overview	3
1.1	Concepts.....	3
1.2	References.....	5
2	Authentication	5
2.1	URL EndPoints	5
2.1.1	Domains:.....	5
2.1.2	URL's:	5
2.1.3	Development	6
2.1.4	Production	6
3	Getting Started.....	6
3.1	Use Postman App	6
3.2	Web Socket HTML Sample	7
4	Basic Service Actions	9
4.1	Flight Plans	10
4.1.1	Create Flight Plan	10
4.1.2	Retrieve.....	13
4.1.3	Update	14
4.1.4	UpdateFlightPlanTimes.....	17
4.1.5	UpdateLocation	18
4.1.6	RetrieveFlightPlansSummary.....	19
4.1.7	Formats for Time Offset.....	20
4.1.8	Time Window.....	21
4.1.9	Vertical and Horizontal Tolerances.....	21
4.1.10	Update Flight Plan Considerations	22
4.2	Alerting.....	23
4.2.1	Evaluate	23
4.2.2	StartTracking.....	26
4.2.3	StopTracking	26
4.2.4	RetrieveAlerts	28
4.3	Briefing	29
4.3.1	AddBrief	29
4.3.2	UpdateBrief.....	31
4.3.3	RetrieveBrief	32

Flight Route Alerting and Briefing – API

5	Receiving Asynchronous alerts	36
5.1	Web Socket	36
5.1.1	URL EndPoints.....	36
5.2	Registering For Additional Web Socket Event Messages	37
5.3	Registering for Refreshed Web Socket Event Message to re-establish state of all tracked flight plans....	37
5.4	Web Socket Ping Pong Usage.....	37
5.5	Web Socket Messages.....	38
6	Example XML.....	39
7	Weather products Tables.....	43
8	Appendix 1. – Flight Plan Alerting Use Case	48
9	Appendix 2. – Error Codes	60
10	Appendix 3. – Info CODES	64
11	Appendix 4. – Demonstration Application.....	66
12	Appendix 5. – Business Rules.....	66
13	Appendix 6. – Implementation Notes.....	67
14	Appendix 7. – Alert Responses.....	68
15	Appendix 8: EFF XML Example	77
16	Document Revisions	82

1 Overview

In this guide, we document the API for Flight Route Alerting and Briefing.

The purpose of this document is to provide all of the information required to our valued customers and partners to take full advantage of DTN’s Flight Route Web Service products.

We will provide the API (Application Programing Interface) features and how to correctly use them within your technical environments. We will also provide the detailed format of the XML message payloads.

1.1 Concepts

Some definitions that are used throughout this document are listed below:

- **Flight Plan** – A definition that identifies the flight as well as the alternate airports and segments that the flight will travel.
- **Briefing** – A summary of weather information that is associated with the Flight Plan. Includes airports and flight segments in ARINC 633-2 standard format

Flight Route Alerting and Briefing – API

- **Tracking** – Monitor the Flight Plan as the weather observations and forecasts are updated and send an alarm if the new weather data is out of specified tolerances.
- **Alarm** – A definition that selects the data of interest and the conditions required in order for an alert to be sent.
- **Alert** – A message that is sent when the conditions associated with an Alarm are satisfied. Note that the first occurrence found of criteria being met triggers an alert. For instance, if an alarm is set for Turbulence greater than 0.3, and a route segment has an area of 0.4 and another area of 0.5 intersecting it, the first found will be returned – 0.4 in many cases – rather than the highest value of 0.5.

1.2 References

XML Schemas:

<http://weather-alerting.dtn.com/xsds/alerting/flightplan/flightPlan.xsd>
<http://weather-alerting.dtn.com/xsds/alerting/flightplan/flightPlansSummary.xsd>
<http://weather-alerting.dtn.com/xsds/alerting/flightplan/flightPlanBrief.xsd>
<http://weather-alerting.dtn.com/xsds/alerting/flightplan/flightPlanAlert.xsd>

Authentication Protocols:

Basic <http://tools.ietf.org/pdf/rfc2617>
OAuth

2 Authentication

Access to this web services requires a username and password.

HTTP Basic is the only method currently supported for authentication.

Note: Even though HTTPS is not strictly enforced for HTTP Basic authentication, it is highly recommended to do so since the password is in a clear text form for HTTP Basic.

Please refer to your account manager to obtain the required credentials for access to the DTN Flight Route Alerting system. If you are experiencing issues with your credentials, you may contact your account manager or aviationsupport@dtm.com for assistance.

2.1 URL EndPoints

All endpoints are available over HTTP and HTTPS. HTTPS is highly recommended.

Note: Our alerting only domains only support alerting features. Our Alerting and Briefing domain support both alerting and briefing capabilities

2.1.1 Domains:

	Test and Dev	Production
Alerting and/or Briefing	c-cat-weather-alerting.dtn.com	c-weather-alerting.dtn.com
Alerting Capabilities Only	cat-weather-alerting.dtn.com	weather-alerting.dtn.com

2.1.2 URL's:

Alerting	/basic/fra
Briefing	/basic/frb

Flight Route Alerting and Briefing – API

2.1.3 Development

For development and acceptance testing of alerting only, please use endpoint <https://cat-weather-alerting.dtn.com/basic/fra>.

For development of Flight Route Alerting and Flight Route briefing features, please use endpoint <https://cat-c-weather-alerting.dtn.com/basic/fra> and <https://cat-c-weather-alerting.dtn.com/basic/frb>

2.1.4 Production

For production of alerting only, please use endpoint <https://weather-alerting.dtn.com/basic/fra>. For development of Flight Route Alerting and Flight Route briefing features, please use endpoint <https://c-weather-alerting.dtn.com/basic/fra> and <https://c-weather-alerting.dtn.com/basic/frb>

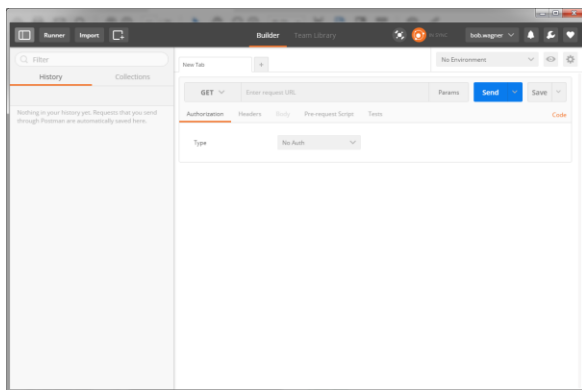
Note: For high availability the DNS record for this server will be updated to transition the product between datacenters. Make sure no DNS caching is being done. If you would like to test network accessibility for referencing other server IPs from your network please contact customer service at aviationsupport@dtm.com.

3 Getting Started

The Flight Route Alerting and Briefing API is a Web Service HTTP API with a web socket Asynchronous communication channel for Alerts. Depending on the actions you are taking you will need to send an HTTP POST or HTTP GET to the Web Service URL endpoint.

3.1 Use Postman App

For the examples in the manual we will be using the Google Chrome Postman app. Postman can be found at the webstore location: <https://chrome.google.com/webstore/search/Postman>.



Flight Route Alerting and Briefing – API

For the examples of web socket connectivity please create a WebSocketClient.html with the following content and open it in Google Chrome.

3.2 Web Socket HTML Sample

```
<!DOCTYPE html>
<html>
  <head>
    <title>DTN Example Websocket Client</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
  </head>
  <body>
    <div>
      User name:<br>
      <input type="text" id="username" name="username"><br>
      User password:<br>
      <input type="password" id="password" name="psw">
    </div>

    <div>
      <br/>
      <input type="text" id="websocketToken">
    </div>

    <div>
      <button type="button" onclick="openSocket();" >Connect</button>
      <button type="button" onclick="closeSocket();" >Disconnect</button>
    </div>
    <!-- Server responses get written here -->
    <div id="messages"></div>

    <!-- Script to utilise the WebSocket -->
    <script type="text/javascript">

      var websocket;
      var messages = document.getElementById("messages");
      var username = document.getElementById("username");
      var password = document.getElementById("password");
      var token;

      function openSocket(){
        // Ensures only one connection is open at a time
        if(websocket !== undefined && websocket.readyState !== WebSocket.CLOSED){
          writeResponse("WebSocket is already opened.");
          return;
        }

        //Get token for the websocket connect call.
        token = requestToken(username, password);

        // Create a new instance of the websocket
        var wsUrlBase = "ws://cat-weather-alerting.dtn.com/basic/frd/websocket?websocketTokenId=";
        websocket = new WebSocket(wsUrlBase.concat(token));

        /**
         * Binds functions to the listeners for the websocket.
         */
        websocket.onopen = function(event){
          if(event.data === undefined)
            return;
          writeResponse(event.data);
        };

        websocket.onmessage = function(event){
          writeResponse(event.data);
        };

        websocket.onclose = function(event){
```

Flight Route Alerting and Briefing – API

```
        writeResponse("Connection closed");
    };

    writeResponse("Connection established.");
}

function requestToken(username, password){
    var xmlHttp = new XMLHttpRequest();
    var token = document.getElementById("websocketToken");
    var basicstr = username.value + ":" + password.value;

    xmlHttp.open( "GET", "https://cat-weather-alerting.dtn.com/basic/fra/websocket/auth", false);
    xmlHttp.setRequestHeader("Content-Type", "text/plain");
    xmlHttp.setRequestHeader("Authorization", "Basic " + btoa(basicstr));
    xmlHttp.send( null );
    return xmlHttp.responseText;
}

function closeSocket(){
    websocket.close();
}

function writeResponse(text){
    messages.innerHTML += "<br/>MESSAGE:<xmp>" + text + "</xmp>";
}
</script>
</body>
</html>
```


Flight Route Alerting and Briefing – API

4 Basic Service Actions

The web services for Flight Routes are an RPC style stateless web service.

The following request types are available:

Create	Submit a new Flight Plan definition to be created in the service.
Retrieve	Get the Flight Plan or Briefing for a specified flight plan that is stored.
Update	Update a stored Flight Plan or Briefing.
UpdateFlightPlanTimes	Update a stored Flight Plan's departure/arrival times.
RetrieveFlightPlansSummary	Gets a list of all flight plan keys, along with the status for each one indicating if it is tracked and if it has active alert.
Evaluate	Evaluate a supplied flight plan and return Alerts.
StartTracking	Start tracking a stored Flight Plan in the service.
StopTracking	Stop tracking a stored Flight Plan in the service.
RetrieveAlerts	Get the current Alert state of a tracked Flight Plan.
AddBrief	Submit a Briefing that is associated with an already created Flight Plan.
UpdateBrief	Trigger the Briefing to be updated with the latest weather information.
RetrieveBrief	Get the Briefing for an associated Flight Plan.

The XML used to create a Flight Plan or Briefing consists of a number of elements. The first components are Airports – an Origin Airport, a Destination Airport, and up to 25 Other Airports. Each of these airports can have a set of Alarms set up on an individual basis.

The next main component of the Flight Plan is the Route. The Route consists of a list of WayPoints, but the Route itself has one set of Alarms for the entire Route.

A more detailed explanation of how the elements in the XML are utilized is in Appendix 1.

4.1 Flight Plans

The flight plan is an XML document written with the [flightPlan.xsd](#)

4.1.1 Create Flight Plan

The Create request is used to input the flight plan information as well as the tolerances of the different aspects to the flight. This request should be used for every unique instance of a flight plan. If you wish to amend an existing flight plan, you should use the Update REQUESTSTYPE that is defined later in this document. The flight plan can be created well in advance of the actual flight as no tracking will be done until the StartTracking request is received.

HTTP POST with Content-Type: application/xml to /basic/fra

Body is a Flight Plan based on:

<http://cat-weather-alerting.dtn.com/xsds/alerting/flightplan/flightPlan.xsd>

The Flight Plan will be identified in our system uniquely based on the following parameters:

<Id>	: Unique Flight Plan identifier provided by customer max 50 characters
<FlightDate>	: This is the UTC Scheduled Flight Date.
<ScheduledDepartureTime>	: This is the UTC Scheduled Departure Time. (Note the Estimated Departure Time can change.)

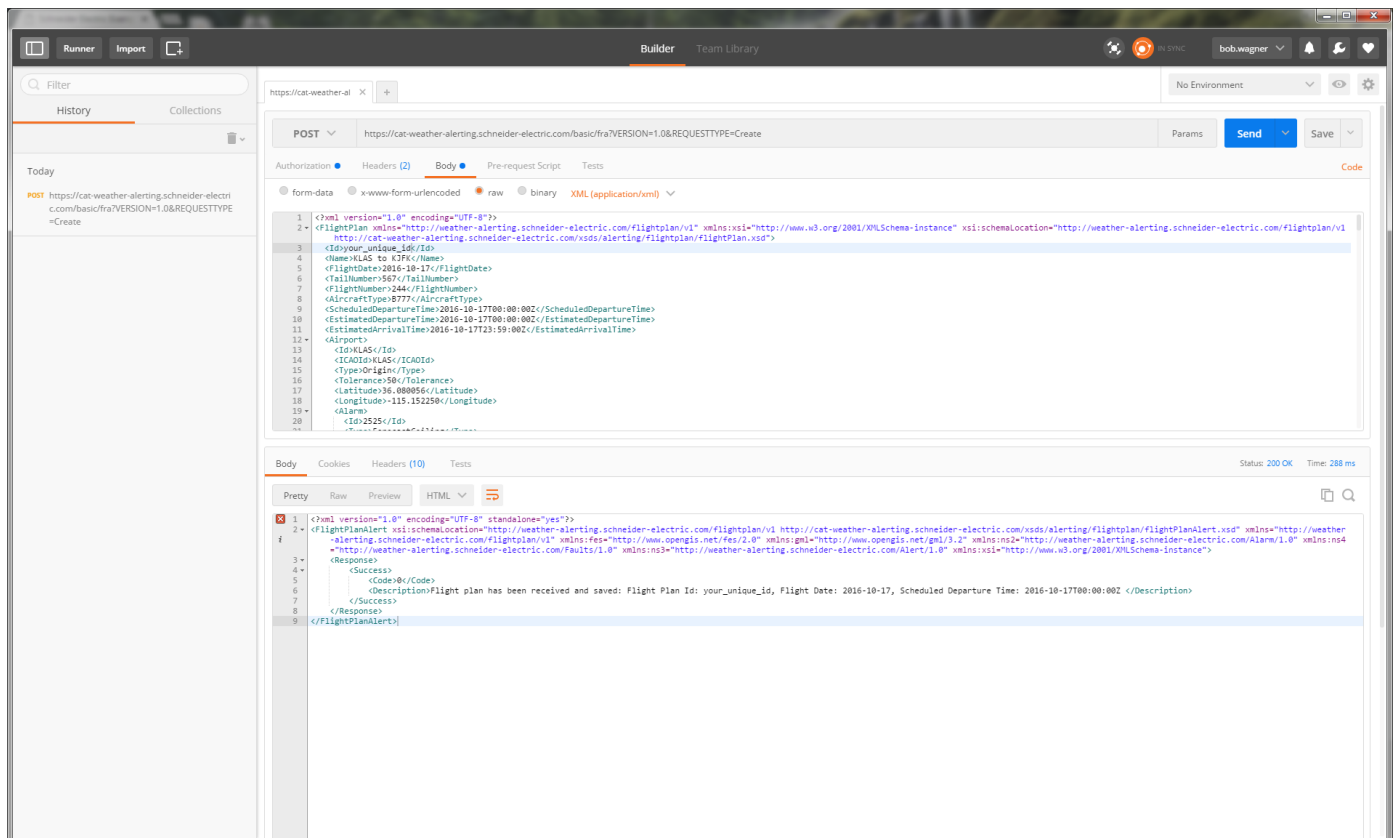
The content-type is "application/xml" and the data in UTF8 format. XML is submitted in the request, as described below.

URL Parameters	Mandatory	Type	Value
VERSION	M	String	"1.0"
REQUESTSTYPE	M	String	"Create"

Flight Route Alerting and Briefing – API

4.1.1.1 Example:

- 1) Open Postman
- 2) From the dropped down for the HTTP request type select POST.
- 3) In the address box, enter the endpoint and parameters `https://cat-weather-alerting.dtn.com/basic/fra?VERSION=1.0&REQUESTTYPE=Create`
- 4) Under the authorization section/tab, pull down the authorization type and select “Basic Auth”.
- 5) Enter your username and password.
- 6) Under the header section/tab, add a Content-Type header with the content “application/xml”.
- 7) Switch to the Body section/tab.
- 8) Using the Flight Plan XML Example [<FlightPlanExample>](#) as the Body of the POST, copy and paste it into the sub-window. Verifying the content type on this screen shows “application/xml”.
- 9) Update the Dates and Times to be the current day.
- 10) Send the request by pressing the Send button.



DTN Confidential Page 12
Updated **June 25, 2020**

Flight Route Alerting and Briefing – API

</Cruise>	O	String	Close of Cruise
<Descent>	O	String	Descent
<Vertical></Vertical>	O	Integer	Vertical tolerance, plus minus from segment, in feet
<Horizontal></Horizontal>	O	Integer	Horizontal tolerance, plus minus from segment, in nautical mi.
</Descent>	O	String	Close of Descent
<TimeBuffer></TimeBuffer>	O	Integer	Minutes to add to segment time range to override time window.
</Tolerance>	O	String	Close of Tolerance
<VerticalTolerance></VerticalTolerance>	M	Integer	Vertical tolerance - DEPRECATED
<HorizontalTolerance></HorizontalTolerance>	O	Integer	Horizontal tolerance - DEPRECATED
<WayPoint>	O	String	Waypoint
<Id></Id>	M	String	Waypoint id, must be unique for flight plan
<Latitude></Latitude>	M	Float	Latitude in decimal degrees
<Longitude></Longitude>	M	Float	Longitude in decimal degrees
<Altitude></Altitude>	M	Integer	Altitude in feet
<TimeOffset></TimeOffset>	M	Integer	Minutes since departure
<Type></Type>	M	String	WayPoint type [Cruise, Climb, Descent], default type is Cruise
</WayPoint>	M	String	Close of Waypoint
<Alarm>	M	String	Alarm
<Id></Id>	M	String	Alarm id, must be unique for flight plan
<Type></Type>	M	String	Alarm type - See Table 1 for available alarm types, properties, values
<Condition>	M	String	Condition
<PropertyIsGreaterThan>	M	String	Operator - See Table 2 for list of available operators
<Name></Name>	M	String	Name of Property
<Value></Value>	M	String	Value of Property
</PropertyIsGreaterThan>	M	String	Close of Operator
</Condition>	M	String	Close of Condition
</Alarm>	M	String	Close of Alarm
</Route>	M	String	Close of Route
</FlightPlan>	M	String	Close of FlightPlan

4.1.2 Retrieve

The Retrieve request is used to read a flight plan that has already been created.

The Retrieve request is a HTTP GET request. URL Parameters are passed in the request, as described below.

In order to retrieve a Flight Plan the user will need to provide the following parameters:

flightPlanId

Flight Route Alerting and Briefing – API

flightDate
scheduledDepartureTime

URL Parameters	Mandatory	Type	
VERSION	M	String	Version - "1.0"
REQUESTTYPE	M	String	Request Type - "Retrieve"
flightPlanId	M	String	Flight plan id, must be unique for customer
flightDate	M	String	Date data type in UTC
scheduledDepartureTime	M	String	Date/Time data type in UTC

4.1.3 Update

The Update request allows the user to update any or all alarm information for a previously submitted flight plan. The user must submit the required flight plan key of the flight plan to update. This key is the flightPlanId, flightDate and the scheduledDepartureTime.

It is important to note that the system will replace ALL alarm criteria for the selected flight plan so the user should submit a list of all alarm criteria. Users may update flight plans in any tracking state, either active or not.

If the flight plan xml is exactly the same as the previously created or updated flight plan xml, then the update will be ignored.

The Update request is a HTTP POST request with content type set to "application/xml" and the data in UTF8 format. XML is submitted in the request, as described below.

URL Parameters	Mandatory	Type	
VERSION	M	String	Version - "1.0"
REQUESTTYPE	M	String	Request Type - "Update"

XML	Mandatory	Type	Description
<?xml version="1.0" encoding="UTF-8"?> <FlightPlan	M	String	FlightPlan

Flight Route Alerting and Briefing – API

<code><Id></Id></code>	M	String	Flight plan id, must be unique for customer
<code><Name></Name></code>	M	String	Name of flight plan
<code><FlightDate></FlightDate></code>	O	String	Date data type in UTC
<code><TailNumber></TailNumber></code>	M	String	Tail number
<code><FlightNumber></FlightNumber></code>	M	String	Flight number
<code><AircraftType></AircraftType></code>	M	String	Aircraft type
<code><ScheduledDepartureTime></ScheduledDepartureTime></code>	M	String	Date/Time data type in UTC
<code><EstimatedDepartureTime></EstimatedDepartureTime></code>	M	String	Date/Time data type in UTC
<code><EstimatedArrivalTime></EstimatedArrivalTime></code>	O	String	Date/Time data type in UTC
<code><StopTrackingLag></StopTrackingLag></code>	M	String	Stop tracking this number of minutes after EstimatedArrivalTime.
<code><Airport></code>	O	Integer	Airport
<code><Id></Id></code>	M	String	Airport id
<code><ICAOId></ICAOId></code>	M	String	Airport ICAO id
<code><Type></Type></code>	O	Integer	Airport type [Origin, Destination, Other]
<code><Tolerance></Tolerance></code>	M	Float	Airport buffer, radius, in nautical mi.
<code><Latitude></Latitude></code>	M	Float	Latitude in decimal degrees
<code><Longitude></Longitude></code>	M	String	Longitude in decimal degrees
<code><Alarm></code>	M	String	Alarm
<code><Id></Id></code>	M	String	Alarm id, must be unique for flight plan
<code><Type></Type></code>	M	String	Alarm type - See Table 1 for available alarm types, properties, values
<code><Condition></code>	M	String	Condition
<code><PropertyIsGreaterThan></code>	M	String	Operator - See Table 2 for list of available operators
<code><Name></Name></code>	M	String	Name of Property
<code><Value></Value></code>	M	String	Value of Property
<code></PropertyIsGreaterThan></code>	M	String	Close of Operator
<code></Condition></code>	M	String	Close of Condition
<code></Alarm></code>	M	String	Close of Alarm
<code></Airport></code>	M	String	Close of Airport
<code><Route></code>	M	String	Route
<code><Tolerance></code>	M	String	Tolerance
<code><Climb></code>	M	String	Climb
<code><Vertical></Vertical></code>	M	String	Vertical tolerance, plus minus from segment, in feet
<code><Horizontal></Horizontal></code>	M	String	Horizontal tolerance, plus minus from segment, in nautical mi.
<code></Climb></code>	M	String	Close of Climb
<code><Cruise></code>	M	String	Cruise
<code><Vertical></Vertical></code>	M	Integer	Vertical tolerance, plus minus from segment, in feet
<code><Horizontal></Horizontal></code>	M	String	Horizontal tolerance, plus minus from segment, in nautical mi.
<code></Cruise></code>	M	String	Close of Cruise
<code><Descent></code>	M	String	Descent
<code><Vertical></Vertical></code>	M	String	Vertical tolerance, plus minus from segment, in feet
<code><Horizontal></Horizontal></code>	M	String	Horizontal tolerance, plus minus from segment, in nautical mi.
<code></Descent></code>	M	String	Close of Descent
<code><TimeBuffer></TimeBuffer></code>	O	Integer	Minutes to add to segment time range to override time window.
<code></Tolerance></code>	M	String	Close of Tolerance

Flight Route Alerting and Briefing – API

<VerticalTolerance></VerticalTolerance>	M	String	Vertical tolerance - DEPRECATED
<HorizontalTolerance></HorizontalTolerance>	M	String	Horizontal tolerance - DEPRECATED
<WayPoint>	M	String	Waypoint
<Id></Id>	M	String	Waypoint id, must be unique for flight plan
<Latitude></Latitude>	M	String	Latitude in decimal degrees
<Longitude></Longitude>	M	String	Longitude in decimal degrees
<Altitude></Altitude>	M	String	Altitude in feet
<TimeOffset></TimeOffset>			Minutes since departure
</WayPoint>			Close of Waypoint
<Alarm>			Alarm
<Id></Id>			Alarm id, must be unique for flight plan
<Type></Type>			Alarm type - See Table 1 for available alarm types, properties, values
<Condition>			Condition
<PropertyIsGreaterThan>			Operator - See Table 2 for list of available operators
<Name></Name>			Name of Property
<Value></Value>			Value of Property
</PropertyIsGreaterThan>			Close of Operator
</Condition>			Close of Condition
</Alarm>			Close of Alarm
</Route>			Close of Route
</FlightPlan>			Close of FlightPlan

Flight Route Alerting and Briefing – API

4.1.4 UpdateFlightPlanTimes

The UpdateFlightPlanTimes request allows the user to update either estimatedDepartureTime, estimatedArrivalTime or both. The user must submit the required flight plan key of the flight plan to update. This key is the flightPlanId, flightDate and the scheduledDepartureTime.

Users may update flight plans in any tracking state, either active or not.

The Update Flight PlanTimes request is a HTTP POST request with content type set to "application/xml" and the data in UTF8 format. XML is not submitted in the request – the payload of the POST is an empty string. URL Parameters are passed in the request, as described below.

URL Parameters	Mandatory	Type	
VERSION	M	String	Version - "1.0"
REQUESTTYPE	M	String	Request Type - "UpdateFlightPlanTimes"
flightPlanId	M	String	Flight plan id, must be unique for customer
flightDate	M	String	Date data type in UTC
scheduledDepartureTime	M	String	Date/Time data type in UTC
estimatedDepartureTime	O	String	Date/Time data type in UTC
estimatedArrivalTime	O	String	Date/Time data type in UTC

Flight Route Alerting and Briefing – API

4.1.5 UpdateLocation

The UpdateLocation is used to identify where the aircraft is currently and when it plans on arriving at the target/destination waypoint. This allows the system to update the time windows of the previous and future waypoints, while identifying which segment is currently active.

In addition, if the current position is provided, it will monitor the flight path to the target waypoint rather than the planned active segment. Reducing and changing what area is monitored for that aircraft until it reaches the target/destination waypoint specified.

The UpdateLocation request is a HTTP POST request with content type set to "application/xml" and the data in UTF8 format. XML is submitted in the request, as described below.

URL Parameters	Mandatory	Type	
VERSION	M	String	Version - "1.0"
REQUESTTYPE	M	String	Request Type - "UpdateLocation"

XML	Mandatory	Type	Description
<?xml version="1.0" encoding="UTF-8"?>			
<LocationUpdate	M	String	LocationUpdate
xmlns="http://weather-alerting.dtn.com/flightplan/v1"			
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"			
xsi:schemaLocation="http://weather-alerting.dtn.com/xsds/alerting/flightplan/locationUpdate.xsd"			
<FlightPlanId></FlightPlanId>	M	String	Flight plan id, must be unique for customer
<FlightDate></FlightDate>	M	String	Date/Time data type in UTC
<ScheduledDepartureTime></ScheduledDepartureTime>	M	String	Date/Time data type in UTC
<TargetWayPoint>	M		
<Id></Id>	M		
<MinutesToArrival></MinutesToArrival>	M	String	Waypoint Id that is defined in the referenced flightplan
</TargetWayPoint>	M	Integer	Minutes until the waypoint is reached from current location
<CurrentPosition>	O		
<Id>CurrentPosition</Id>	O	String	Must be "CurrentPosition"
<Latitude></Latitude>	O	Float	Latitude in decimal degrees
<Longitude></Longitude>	O	Float	Longitude in decimal degrees
<Altitude></Altitude>	O	Integer	Altitude in feet
</CurrentPosition>	O		
</LocationUpdate>	M		

Response: If the system is unable to update the flightplan it will return with an HTTP error and response payload. Else, it will return an alert with acknowledgement of command.

Flight Route Alerting and Briefing – API

As an alternative for the MinutesToArrival parameter, it is possible to specify the actual UTC time of arrival at the target waypoint as follows:

```
<TargetWayPoint>  
  <Id>...</Id>  
  <ArrivalTime>YYYY-MM-DDTHH:MM:SSZ</ArrivalTime>  
</TargetWayPoint>
```

4.1.6 RetrieveFlightPlansSummary

The RetrieveFlightPlansSummary request is used to retrieve a list of all flight plan keys, along with the status for each one indicating if it is tracked and if it has active alerts. Only returns information for flight plans created in the last maxAge hours

The RetrieveFlightPlansSummary request is a HTTP GET request. URL Parameters are passed in the request, as described below.

URL Parameters	Mandatory	Type	
VERSION	M	String	Version - "1.0"
REQUESTTYPE	M	String	Request Type - "RetrieveFlightPlansSummary"
maxAge	O	Integer	Maximum age in hours, default is 24

This is the response.

RetrieveFlightPlansSummary Response XML	
<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <FlightPlanSummary xsi:schemaLocation="http://weather-alerting.dtn.com/flightplan/v1 http://cat-weather-alerting.dtn.com/xsds/alerting/flightplan/flightPlansSummary.xsd" xmlns="http://weather-alerting.dtn.com/flightplan/v1" xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:ns2="http://weather-alerting.dtn.com/Alarm/1.0" xmlns:ns4="http://weather-alerting.dtn.com/Faults/1.0" xmlns:ns3="http://weather-alerting.dtn.com/Alert/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></pre>	FlightPlanSummary

Flight Route Alerting and Briefing – API

<pre><FlightPlan> <Id>1443811965767</Id> <FlightDate>2015-10-02</FlightDate> <TailNumber>567</TailNumber> <ScheduledDepartureTime>2015-10-02T19:52:00Z</ScheduledDepartureTime> <CreationTime>2015-10-02T18:52:46Z</CreationTime> <IsPending>false</IsPending> <IsTracked>true</IsTracked> <HasActiveAlerts>true</HasActiveAlerts> </FlightPlan> </FlightPlanSummary></pre>	FlightPlan Close of FlightPlan Close of FlightPlanSummary
---	---

4.1.7 Formats for Time Offset

Most of the examples throughout this document show the TimeOffset parameter to declare, for each waypoint, the number of minutes from the estimated departure time. However, there are actually other alternatives to declare the time at the waypoint.

Time at the waypoint can be declared as a difference in minutes between this waypoint and the previous one:

```
<WayPoint>
  ... other waypoint parameters ...
  <TimeDifference>NN</TimeDifference>
</WayPoint>
```

Time at the waypoint can be declared as UTC date/time as follows:

```
<WayPoint>
  ... other waypoint parameters ...
  <TimeStamp>YYYY-MM-DDTHH:MM:SSZ</TimeStamp>
</WayPoint>
```

The three different formats (TimeOffset, TimeDifference, TimeStamp) are mutually exclusive in a given waypoint, and they are expected to declare a time after the estimated departure time.

4.1.8 Time Window

When evaluating a flight plan for weather alerts, FRA creates a time window for each segment and looks for an overlap with the time range of a weather event. The default time window is defined by the start and end time of the segment; however, this default time window can be overridden by the definitions in the flight plan.

The optional <TimeBuffer> tolerance parameter helps to adjust the size of the time window.

```
<Tolerance>  
  ... other tolerance settings ...  
  <TimeBuffer>NN</TimeBuffer>  
</Tolerance>
```

Where NN is the time in minutes to subtract from the start time of the segment and to add to the end time of the segment to create the time window.

This parameter is completely optional and it may be of some help for those cases where some variation is expected for the actual times of the route segments. However, the alternative of just submitting an updated flight plan when times change should be considered instead of expanding the time window.

4.1.9 Vertical and Horizontal Tolerances

There is a limit to the extent of the vertical and horizontal tolerances for different phases of the flight. Please note the table below which details the maximum tolerances, plus or minus from the segment.

	Horizontal Tolerance Maximum	Vertical Tolerance Maximum
Climb	200nm	--
Cruise	200nm	5000 feet
Descent	200nm	--

4.1.10 Update Flight Plan Considerations

It is important to note that those requests that successfully update the flight plan, always present the following behavior:

- 1) A generic <FlightPlanAlert> message is returned to HTTP client with the status of the request.
- 2) A <FlightPlanUpdate> message is sent to the websocket clients.
- 3) If the flight pan was already being tracked, an updated <FlightPlanAlert> message is sent to the websocket clients.

Since the update operation resets the state of all the alerts, for an active flight plan, the client side of FRA must take an equivalent action if it is keeping an internal memory of the alerts.

The requests that update the flight plan are: Update, UpdateFlightPlanTimes, and UpdateLocation.

Flight Route Alerting and Briefing – API

4.2 Alerting

4.2.1 Evaluate

The Evaluate request is used to create a flight plan, run a one-time check of all of the alarms against the flight plan. The response of this request lists any active alerts corresponding to the alarms that are set. It is important to note that the system will not add this flight plan to the active monitoring list. If the user wants to track this flight plan, it should be provided via the “Create” event.

This is a HTTP POST request with content type set to "application/xml" and the data in UTF8 format. URL Parameters are passed in the request, as described below.

URL Parameters	
VERSION	string, "1.0"
REQUESTTYPE	string, "Evaluate"

XML	Mandatory	Type	Description
<pre><?xml version="1.0" encoding="UTF-8"?> <FlightPlan xmlns="http://weather-alerting.dtn.com/flightplan/v1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://cat-weather- alerting.dtn.com/xsds/alerting/flightplan/flightPlanAlert.xsd"> <Id></Id> <Name></Name> <FlightDate></FlightDate> <TailNumber></TailNumber> <FlightNumber></FlightNumber> <AircraftType></AircraftType> <ScheduledDepartureTime></ScheduledDepartureTime> <EstimatedDepartureTime></EstimatedDepartureTime> <EstimatedArrivalTime></EstimatedArrivalTime> <Airport> <Id> </Id> <ICAOId> </ICAOId> <Type> </Type> <Tolerance></Tolerance></pre>	M	String	FlightPlan
	M	String	Flight plan id, must be unique for customer
	O	String	Name of flight plan
	M	String	Date data type in UTC
	M	String	Tail number
	M	String	Flight number
	M	String	Aircraft type
	M	String	Date/Time data type in UTC
	M	String	Date/Time data type in UTC
	O	String	Date/Time data type in UTC
	M	String	Airport
	M	String	Airport id
	M	String	Airport ICAO id
	M	String	Airport type [Origin, Destination, Other]
	M	Integer	Airport buffer, radius, in nautical mi.

Flight Route Alerting and Briefing – API

<Latitude></Latitude>	M	Float	Latitude in decimal degrees
<Longitude></Longitude>	M	Float	Longitude in decimal degrees
<Alarm>	M	String	Alarm
<Id></Id>	M	String	Alarm id, must be unique for flight plan
<Type></Type>	M	String	Alarm type - See Table 1 for available alarm types, properties, values
<Condition>	M	String	Condition
<PropertyIsGreaterThan>	M	String	Operator - See Table 2 for list of available operators
<Name></Name>	M	String	Name of Property
<Value></Value>	M	String	Value of Property
</PropertyIsGreaterThan>	M	String	Close of Operator
</Condition>	M	String	Close of Condition
</Alarm>	M	String	Close of Alarm
</Airport>	M	String	Close of Airport
<Route>	M	String	Route
<Tolerance>	M	String	Tolerance
<Climb>	M	String	Climb
<Vertical></Vertical>	M	String	Vertical tolerance, plus minus from segment, in feet
<Horizontal></Horizontal>	M	String	Horizontal tolerance, plus minus from segment, in nautical mi.
</Climb>	M	String	Close of Climb
<Cruise>	M	String	Cruise
<Vertical></Vertical>	M	String	Vertical tolerance, plus minus from segment, in feet
<Horizontal></Horizontal>	M	Integer	Horizontal tolerance, plus minus from segment, in nautical mi.
</Cruise>	M	String	Close of Cruise
<Descent>	M	String	Descent
<Vertical></Vertical>	M	String	Vertical tolerance, plus minus from segment, in feet
<Horizontal></Horizontal>	M	String	Horizontal tolerance, plus minus from segment, in nautical mi.
</Descent>	M	String	Close of Descent
<TimeBuffer></TimeBuffer>	M	Integer	Minutes to add to segment time range to override time window.
</Tolerance>	M	String	Close of Tolerance
<VerticalTolerance></VerticalTolerance>	M	String	Vertical tolerance - DEPRECATED
<HorizontalTolerance></HorizontalTolerance>	M	String	Horizontal tolerance - DEPRECATED
<WayPoint>	M	String	Waypoint
<Id></Id>	M	String	Waypoint id, must be unique for flight plan
<Latitude></Latitude>	M	String	Latitude in decimal degrees
<Longitude></Longitude>	M	String	Longitude in decimal degrees
<Altitude></Altitude>	M	String	Altitude in feet
<TimeOffset></TimeOffset>			Minutes since departure
</WayPoint>			Close of Waypoint
<Alarm>			Alarm
<Id></Id>			Alarm id, must be unique for flight plan
<Type></Type>			Alarm type - See Table 1 for available alarm types, properties, values
<Condition>			Condition
<PropertyIsGreaterThan>			Operator - See Table 2 for list of available operators
<Name></Name>			Name of Property
<Value></Value>			Value of Property
</PropertyIsGreaterThan>			Close of Operator
</Condition>			Close of Condition
</Alarm>			Close of Alarm

Flight Route Alerting and Briefing – API

</Route> </FlightPlan>			Close of Route Close of FlightPlan
---------------------------	--	--	---------------------------------------

This is the response.

Evaluate Response XML	
<pre> <?xml version="1.0" encoding="UTF-8"?> <FlightPlanAlert xsi:schemaLocation="http://cat-weather- alerting.dtn.com/xsds/alerting/flightplan/flightPlanAlert.xsd" xmlns="http://weather-alerting.dtn.com/flightplan/v1" xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:ns2="http://weather-alerting.dtn.com/Alarm/1.0" xmlns:ns4="http://weather-alerting.dtn.com/Faults/1.0" xmlns:ns3="http://weather-alerting.dtn.com/Alert/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <FlightPlanId>47712093471023</FlightPlanId> <FlightDate>2014-12-25</FlightDate> <TailNumber>567</TailNumber> <FlightNumber>244</FlightNumber> <ScheduledDepartureTime>2014-12-25T10:00:00Z</ScheduledDepartureTime> <OriginAirportId>KLAS</OriginAirportId> <DestinationAirportId>KJFK</DestinationAirportId> <Airport> <Id> </Id> <Type> </Type> <Alert> <AlarmId></AlarmId> <Time></Time> <Data> <Property> <Name> </Name> <Value></Value> </Property> </Data> </Alert> </Airport> <Route> <Segment> <WayPointId></WayPointId> <WayPointId></WayPointId> <Alert> <AlarmId></AlarmId> <AlarmType></AlarmType> <Time></Time> </pre>	FlightPlanAlert Flight plan id Date data type in UTC Tail number Flight number Date/Time data type in UTC Origin airport id Destination airport id Airport Airport id Airport type Alert Alarm id for the alarm that triggered Time the alarm triggered Data Property Property of the alarm that triggered Actual value that triggered the alarm Close of Property Close of Data Close of Alert Close of Airport Route Segment First waypoint id Second waypoint id Alert Alarm id for the alarm that triggered Alarm type Time the alarm triggered

Flight Route Alerting and Briefing – API

<pre><Data> <Property> <Name></Name> <Value></Value> </Property> </Data> </Alert> </Segment> </Route> </FlightPlanAlert></pre>	Data Property Property of the alarm that triggered Actual value that triggered the alarm Close of Property Close of Data Close of Alert Close of Segment Close of Route Close of FlightPlanAlert
--	---

4.2.2 StartTracking

The Start Tracking request is used to begin tracking a previously created flight plan on a five minute interval. Every five minutes the flight plan will be evaluated against all alarm criteria specified by the user. A message will be sent on the connected web socket for all active alerts (New, Continuing and Cleared).

This is a HTTP POST request with content type set to "application/xml" and the data in UTF8 format. Note that the message body is empty. URL Parameters are passed in the request, as described below.

URL Parameters	Mandatory	Type	
VERSION	M	String	Version - "1.0"
REQUESTTYPE	M	String	Request Type - "StartTracking"
flightPlanId	M	String	Flight plan id, must be unique for customer
flightDate	M	String	Date data type in UTC
scheduledDepartureTime	M	String	Date/Time data type in UTC

4.2.3 StopTracking

Flight Route Alerting and Briefing – API

The Stop Tracking request is used to end the tracking of a flight plan. Alert messages will no longer be sent on the connected web socket for the flight plan. Once the system receives the STOP TRACKING request, the flight plan is removed from the active tracking list and no further actions will be taken.

This is a HTTP POST request with content type set to "application/xml" and the data in UTF8 format. Note that the message body is empty. URL Parameters are passed in the request, as described below.

URL Parameters	Mandatory	Type	
VERSION	M	String	Version - "1.0"
REQUESTTYPE	M	String	Request Type - "StopTracking"
flightPlanId	M	String	Flight plan id, must be unique for customer
flightDate	M	String	Date data type in UTC
scheduledDepartureTime	M	String	Date/Time data type in UTC

Please also note that in the case where no "StopTracking" request is received for a flight plan, the system will automatically issue a StopTracking, putting the flight plan into an untracked state, right after the ETA (Estimated Time of Arrival) is reached.

In the case the ETA is not provided, the system will automatically issue a StopTracking at a time based on the given end time at airports and/or the time at the last waypoint point of the route.

To protect system resources, a flight plan is not tracked for more than 5 days.

4.2.4 RetrieveAlerts

The RetrieveAlerts request is used to retrieve the active alerts for a flight plan that has already been created and is being tracked.

The RetrieveAlerts request is a HTTP GET request. URL Parameters are passed in the request, as described below.

URL Parameters	Mandatory	Type	
VERSION	M	String	Version - "1.0"
REQUESTTYPE	M	String	Request Type - "RetrieveAlerts"
flightPlanId	M	String	Flight plan id, must be unique for customer
flightDate	M	String	Date data type in UTC
scheduledDepartureTime	M	String	Date/Time data type in UTC

4.3 Briefing

4.3.1 AddBrief

Associate a Brief with a specified Flight Plan that has already been created.

Airports

Identify which airports you wish to have a briefing for. (The Id provided in the flight Plan Airport Section.)

Note: Airport Maps will have (Radar, Lightning, METAR, , Forecast Ceiling, and Forecast Visibility.)

Note: The desired radius around the airports will be taken from the Flight Plan tolerance for each airport.

Flight Legs

Define the time window for breaking apart the flight route into Legs. This can be 1, 2 or 3 hour legs.

Note

Legs for a briefing are different than segments for an alert. An alert segment is between two waypoints.

A briefing leg is time bound. This means a leg may have multiple waypoints within it.

Identify the format(s) you want the briefing available in? (633 only to start with)

Identify the Maps Layer Group they want for each segment. (Hazards, , Significant Weather and US Only)

Hazards Layer Group includes the following layers:

- Turbulence – North American (NAM) and Global
- Icing – NAM and Global
- Thunderstorms – NAM and Global

Significant Weather Layer Group includes the following layers:

- Winds Aloft
- Jet Streams
- Tropopause Heights
- Tropopause Lows
- Volcanic Ash
- Volcanic Advisory
- SIGMETs
- Fronts and Pressure Centers

Flight Route Alerting and Briefing – API

US Only Layer Group includes the following layers:

- TFR
- PIREP
- Icing AIRMET
- Turbulence AIRMET
- IFR/ Mobs AIRMET
- Convective SIGMET

This is a HTTP POST request with content type set to "application/xml" and the data in UTF8 format.

URL Parameters

VERSION
REQUESTTYPE

string, "1.0"
string, "AddBrief"

XML	Mandatory	Type	Description
<?xml version="1.0" encoding="UTF-8"?> <FlightPlanBrief xmlns="http://weather-alerting.dtn.com/flightplan/v1" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <FlightPlanRef> <Id>Identifier123</Id> <FlightDate>2015-07-14</FlightDate> <ScheduledDepartureTime> 2015-06-23T15:00:00Z </ScheduledDepartureTime> </FlightPlanRef> <AirportBrief> <Id>KSTP</Id> </AirportBrief> <AirportBrief> <Id>PHOG</Id> </AirportBrief> <RouteBrief>	M	String	FlightPlanBrief
<FlightPlanRef>	M	String	Reference to an already saved Flight Plan
<Id>Identifier123</Id>	M	String	Customer identifier for the flight plan
<FlightDate>2015-07-14</FlightDate>	M	String	Flight Date data type in UTC format YYYY-MM-DD
<ScheduledDepartureTime> 2015-06-23T15:00:00Z	M	String	Scheduled Date/Time data type in UTC format YYYY-MM-DDTHH:MM:ssZ
</ScheduledDepartureTime>			
</FlightPlanRef>			
<AirportBrief>	M	Entity	Airport
<Id>KSTP</Id>	M	String	Airport ICAO id - must match an airport provided in the Flight Plan
</AirportBrief>			
<AirportBrief>	M	Entity	Airport
<Id>PHOG</Id>	M	String	Airport ICAO id - must match an airport provided in the Flight Plan
</AirportBrief>			
<RouteBrief>	M	Entity	RouteBrief

Flight Route Alerting and Briefing – API

<TimeWindow>1</TimeWindow>	M	Integer	The length of time for each flight segment in hours. Min of 1, max of 3
<Format>633-2</Format>	O	String	Format of response – currently only 633-2 supported
<RouteMap>FlightHazards</RouteMap>	M	String	One or more RouteMap types. This defines the layers for each flight segment. See above for details
</RouteBrief>	M	String	
</FlightPlanBrief>			

This is the response.

Evaluate Response XML	
<pre><?xml version="1.0" encoding="UTF-8" standalone="yes" ?> <FlightPlanAlert xsi:schemaLocation="http://weather-alerting.dtn.com/flightplan/v1 http://az-fra-web-pre01/xsds/alerting/flightplan/flightPlanAlert.xsd"> <Response> <Success> <Code>0</Code> <Description>A Flight Brief has been saved for: Flight Plan Id: Identifier123, Flight Date: 2015-07-14, Scheduled Departure Time: 2015-06-23T15:00:00Z </Description> </Success> </Response> </FlightPlanAlert></pre>	FlightPlanAlert Response Success – identifying that request was received Code – 0 for success, see error code schedule for errors Description – of flight plan for which brief was added

4.3.2 UpdateBrief

Have the brief updated with the latest weather data.

This request is very similar to the AddBrief, except it is using the same FlightPlan and information as from a previous brief. All flight plan details are the same, but since time has changed, there may be new weather updates. This allows us to get an updated brief with updated weather. You can also specify different airports, time window or RouteMap values.

This is a HTTP POST request with content type set to "application/xml" and the data in UTF8 format.

Flight Route Alerting and Briefing – API

URL Parameters

VERSION	string, "1.0"
REQUESTTYPE	string, "UpdateBrief"
flightPlanId	The flightPlanId of the flight plan for which and updated is requested.
flightDate	The flight date in UTC of the flight plan for which and updated is requested.
scheduledDepartureTime	The scheduled departure time in UTC of the flight plan for which and updated is requested.

The input XML is the same as for AddBrief

4.3.3 RetrieveBrief

Retrieve a Brief with a specified Flight Plan that has already been added.

Identify the format you want the briefing in? And section.

633 EFF, 633 Airport Weather

The ARINC 633-2 EFF file will be a zip file of all airport, flight segment images with weather layers, along with the ARINC 633-2 AirportWeather.xml containing textual version of METAR and TAF data for each of the airports.

To retrieve the ARINC 633-2 EFF

This will be a HTTPS GET request

URL Parameters	Mandatory	Type	
VERSION	M	String	Version - "1.0"
REQUESTTYPE	M	String	Request Type - "RetrieveBrief"
flightPlanId	M	String	Flight plan id, must be unique for customer
flightDate	M	String	Date data type in UTC
scheduledDepartureTime	M	String	Date/Time data type in UTC
Header Parameters			
Accept: application/zip			

Response: ARINC 633-2 formatted zip file.

Flight Route Alerting and Briefing – API

To Retrieve the ARINC 633-2 AirportWeather.xml only – note, it is included in the EFF, but can be retrieved independently

This will be a HTTPS GET request

URL Parameters	Mandatory	Type	
VERSION	M	String	Version - "1.0"
REQUESTTYPE	M	String	Request Type - "RetrieveBrief"
onlyAirportWeather	M	Boolean	Set to true to get AirportWeather only. =true.
flightPlanId	M	String	Flight plan id, must be unique for customer
flightDate	M	String	Date data type in UTC
scheduledDepartureTime	M	String	Date/Time data type in UTC
Header Parameters			
Accept: application/xml			

The above to options provide for ARINC 633-2 compliant responses. However, there are options to just obtain certain portions of the brief response, such as only airport images or just flight route images, although this is more involved.

You need to make a request to get the EFF.xml file. This XML file contains descriptions and identifiers for each of the images.

To get the EFF.xml file – this is a HTTPS GET request – the parameters are exactly the same as for the zip file, but the Accept header is application/xml instead of application/zip.

EFF XML Output – See full example in Appendix 15 Appendix 8: EFF XML Example

Element	Description
EFUSUB	Primary Root Element
M633Header	Various official Header information including version number and timestamp of creation
Flight	Information about the flight as provided in the flightplan, including airports, arrival and departure times
Aircraft	Information about the aircraft as provided in the flightplan including registration and model
SubFolder title="_Root_Folder_"	This section now breaks out a logical folder structure and the documents that are in each folder. The documents and images

Flight Route Alerting and Briefing – API

	themselves are all stored at the root level, this SubFolder structure allows for a logical grouping of the documents with a reference to the documents within the folder.
SubFolder title="Airport Weather"	This folder will logically group all the documents with airport related weather.
SubFolder title="Flight Weather"	This folder will logically group all the documents related to the flight route
Document	Description of each document with the attributes providing information about the document:
title="KSGR - Sugar Land Regional Airport, United States"	A descriptive title of the document. Airports – style ICAO Id – Airport Name Flight Route – Leg from start minute to end minute
id="t-f15dc50b-b06d-43a9-8ec3-f77731fad5a4"	A UUID identifier for the document. This identifier can be used in
file="KSGR_-_Sugar_Land_Regional_Airport,_United_States.png"	A name for the file, often having the file extension to help identify type, such as .png. Usually similar to title.
updateDateTime="2017-01-31T16:01:56.744Z"	Date and time document was created in UTC
application="DTN"	Application will always be DTN
authorProfile="dispatcher"	authorProfile will always be dispatcher
<Topic name="Weather/Airport"/>	Defines this as a Weather and Airport related topic from 633

To get just a single document – note, you need to get the EFF.xml first and use the document identifiers for the next call

This is a HTTPS GET Request

Note, Accept header is important. If the resource type is an image, the Accept header is application/png. If the resource type is XML, the Accept header is application/xml.

Flight Route Alerting and Briefing – API

URL Parameters	
VERSION	string, "1.0"
REQUESTTYPE	string, "RetrieveBrief"
resourceId	string, UUID identifier from document of eff.xml
Headers	
Accept: application/xml	For .xml type documents
Accept: application/png	For .png type documents

Summary of GET Brief Requests

Attributes and Headers	Description
&REQUESTTYPE=RetrieveBrief	Required request type for all retrieval of brief artifacts is 'RetrieveBrief'
&version=1.0	Optional version – defaults to 1.0
Accept: application/zip	Used to get the ARINC 633-2 EFF zip file
Accept: application/xml	Used to when getting the AirportWeather.xml, EFF.xml or briefing resources that are XML documents
Authentication: Basic xxxxx	Required for authentication of all requests
&flightPlanId	FlightPlan Identifier provided in the Flight Plan creation – used for all requests except resource requests
&flightDate	Date of the flight in UTC – used for all requests except resource requests
&scheduledDepartureTime	Scheduled departure date and time of the flight in UTC as provided in the flight plan – used for all requests except resource requests
&onlyAirportWeather=true	Supplied when the ARINC 633-2 AirportWeather.xml is requested
&resourceId	The identifier as obtained from the EFF.xml as the id attribute of the Document element.

Flight Route Alerting and Briefing – API

5 Receiving Asynchronous alerts

5.1 Web Socket

5.1.1 URL EndPoints

Alerting responses are received over WS (WebSocket) and WSS (WebSocket over TLS). Access to this web socket requires a token, which is requested by a HTTP GET request with content type set to "text/plain" with HTTP Basic authentication, and is passed in on the connection as a query parameter websocketTokenId=<token>.

Domains:

	Test and Dev	Production
Alerting and Briefing	c-cat-weather-alerting.dtn.com	c-weather-alerting.dtn.com
Existing Alerting Only customers	cat-weather-alerting.dtn.com	weather-alerting.dtn.com

URL

	Protocol	URL
Get Authentication Token	https	/basic/fra/websocket/auth
Connect to Web Socket	wss	/basic/frd/websocket?websocketTokenId=<token>

Note: For high availability the DNS record for this server will be updated to transition the product between datacenters. Make sure no DNS caching is being done. If you would like to test network accessibility for referencing other server IPs from your network please contact customer service at aviationsupport@dtm.com.

Flight Route Alerting and Briefing – API

5.2 Registering For Additional Web Socket Event Messages

By default, Alert events are sent as messages on the web socket. Other events are available as well – Created, Updated, Tracked and Untracked. To register to receive these event messages, pass in an optional parameter – Stream – as follows:

```
ws://cat-weather-  
alerting.dtn.com/basic/frd/websocket?websocketTokenId=<mytoken>&Stream=FlightPlanAlert&Stream=FlightPlanCreated&Stream=FlightPlanU  
pdated&Stream=FlightPlanTracked&Stream=FlightPlanUntracked
```

5.3 Registering for Refreshed Web Socket Event Message to re-establish state of all tracked flight plans.

By default, Alert events are sent as messages on the web socket for new or tracked messages with current state information. As the state of a flight transitions through new, continuing and cleared alerts, it may be needed to refresh the clients understanding of the current state due to a socket disconnection or other reason. In order to receive a message that will represent the current active state on the next update cycle, use the 'refresh' parameter and set it to 'true'. By default, it is 'false'.

Example: `ws://cat-weather-alerting.dtn.com/basic/frd/websocket?websocketTokenId=<mytoken>
refresh=true&Stream=FlightPlanAlert&Stream=FlightPlanCreated&Stream=FlightPlanUpdated&Stream=FlightPlanTracked&Stream=FlightPlanUn
tracked`

You will receive 1 message that will be a complete state of all active alerts for the next evaluation cycle for each flight plan that was being tracked. After which it will be a normal state event message.

5.4 Web Socket Ping Pong Usage

Due to the nature of the web socket communication, the presence of a variety of elements like proxies, firewalls, NATs, etc. is expected between the client and the server. Each of them with their own idle timer and rules, which could cause a disconnection for lack of activity.

In the worst of the cases, that disconnection could be produced without the client nor the server receiving a formal indication, which could leave the client waiting for an alert without knowing that the connection was already dropped. In those cases where the client sends messages to the server, they can function as a probe that would reveal a dropped connection; however, in our case, the messages are mostly from the server to the client.

Flight Route Alerting and Briefing – API

To avoid this situation, it is common to use the ping pong mechanism, which helps to keep the connection alive and receive a prompt indication when the connection has been dropped. This mechanism is part of the Websocket Protocol (RFC6455) and, put in simple terms, consists of sending a periodic ping signal to the server and expecting a pong in response. The pong response confirms that the connection is alive.

Depending on the web socket framework used on the client side, it may be as simple as setting a parameter when starting the listener.

In our specific case, a **ping interval of 5 minutes** is recommended as a starting point but adjustments may be done if needed.

5.5 Web Socket Messages

The system will maintain and report back to the users a list of active alerts for each flight plan submitted with the status of each alarm. The following are the possible statuses each alert may have:

New : Indicates that this is the first instance of the alarm generated for the flight plan
Continuing : Indicates that the alarm is ongoing and has already been reported as new previously
Cleared : Indicates that this alarm condition no longer exists for the flight plan and will no longer be reported

The example below illustrates what the users can expect as part of the Alert stream they will receive via the web socket connection.

XML	
<pre><?xml version="1.0" encoding="UTF-8"?> <FlightPlanAlert xmlns="http://www.dtn.com/weather/alerting/v1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://cat-weather- alerting.dtn.com/xsds/alerting/flightplan/flightPlanAlert.xsd"> <FlightPlanId>47712093471023</FlightPlanId> <FlightDate>2014-12-25</FlightDate> <TailNumber>567</TailNumber> <FlightNumber>244</FlightNumber> <ScheduledDepartureTime>2014-12-25T10:00:00Z</ScheduledDepartureTime> <OriginAirportId>KLAS</OriginAirportId> <DestinationAirportId>KJFK</DestinationAirportId> <Airport> <Id> </Id> <Type> </Type> </Airport> <Alert> <AlarmId></AlarmId> <Time></Time></pre>	FlightPlanAlert Flight plan id Date data type in UTC Tail number Flight number Date/Time data type in UTC Origin airport id Destination airport id Airport Airport id Airport type Alert Alarm id for the alarm that triggered Time the alarm triggered

Flight Route Alerting and Briefing – API

<pre> <Data> <Property> <Name> </Name> <Value> </Value> </Property> </Data> </Alert> </Airport> <Route> <Segment> <WayPointId> </WayPointId> <WayPointId> </WayPointId> <Alert> <AlarmId> </AlarmId> <AlarmType> </AlarmType> <Status> </Statue> <Time> </Time> <Data> <Property> <Name> </Name> <Value> </Value> </Property> </Data> </Alert> </Segment> </Route> </FlightPlanAlert> </pre>	<pre> Data Property Property of the alarm that triggered Actual value that triggered the alarm Close of Property Close of Data Close of Alert Close of Airport Route Segment First waypoint id Second waypoint id Alert Alarm id for the alarm that triggered Alarm type Status – New, Continuing, Cleared Time the alarm triggered Data Property Property of the alarm that triggered Actual value that triggered the alarm Close of Property Close of Data Close of Alert Close of Segment Close of Route Close of FlightPlanAlert </pre>
--	---

6 Example XML

Below is an example of a complete Create or Evaluate request XML structure that can be used by customers to submit flight plans to the Flight Route Alerting system.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<FlightPlan xmlns="http://weather-alerting.dtn.com/flightplan/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://weather-alerting.dtn.com/flightplan/v1 http://cat-weather-alerting.dtn.com/xsds/alerting/flightplan/flightPlan.xsd">
  <Id>your_unique_id</Id>
  <Name>EGLL to EDDF</Name>
  <FlightDate>2017-04-17</FlightDate>
  <TailNumber>567</TailNumber>

```

Flight Route Alerting and Briefing – API

```
<FlightNumber>244</FlightNumber>
<AircraftType>B777</AircraftType>
<ScheduledDepartureTime>2017-04-20T02:15:38Z</ScheduledDepartureTime>
<EstimatedDepartureTime>2017-04-20T02:15:38Z</EstimatedDepartureTime>
<EstimatedArrivalTime>2017-04-20T03:05:38Z</EstimatedArrivalTime>
<Airport>
  <Id>EGLL</Id>
  <ICAOId>EGLL</ICAOId>
  <Type>Origin</Type>
  <Tolerance>10</Tolerance>
  <Latitude>51.469722</Latitude>
  <Longitude>-0.452222</Longitude>
  <Alarm>
    <Id>2525</Id>
    <Type>ForecastCeiling</Type>
    <Condition>
      <PropertyIsGreaterThan>
        <Name>Altitude</Name>
        <Value>10000</Value>
      </PropertyIsGreaterThan>
    </Condition>
  </Alarm>
</Airport>
<Airport>
  <Id>EDDF</Id>
  <ICAOId>EDDF</ICAOId>
  <Type>Destination</Type>
  <Tolerance>10</Tolerance>
  <Latitude>50.050000</Latitude>
  <Longitude>8.600000</Longitude>
  <Alarm>
    <Id>2528</Id>
    <Type>ForecastCeiling</Type>
    <Condition>
      <PropertyIsGreaterThan>
        <Name>Altitude</Name>
        <Value>10000</Value>
      </PropertyIsGreaterThan>
    </Condition>
  </Alarm>
<Alarm>
  <Id>2529</Id>
  <Type>Lightning</Type>
  <Condition>
    <Exists/>
  </Condition>
</Alarm>
<Alarm>
```


Flight Route Alerting and Briefing – API

```
<Id>2530</Id>
<Type>TAF</Type>
<Condition>
  <Or>
    <PropertyIsEqualTo>
      <Name>FlightCategory</Name>
      <Value>IFR</Value>
    </PropertyIsEqualTo>
    <PropertyIsEqualTo>
      <Name>FlightCategory</Name>
      <Value>LIFR</Value>
    </PropertyIsEqualTo>
  </Or>
</Condition>
</Alarm>
</Airport>
<Route>
<Tolerance>
  <Vertical>1000</Vertical>
  <Horizontal>50</Horizontal>
  <TimeBuffer>5</TimeBuffer>
</Tolerance>
<WayPoint>
  <Id>1</Id>
  <Latitude>36.53000</Latitude>
  <Longitude>-114.61000</Longitude>
  <Altitude>16000</Altitude>
  <TimeOffset>20</TimeOffset>
</WayPoint>
<WayPoint>
  <Id>2</Id>
  <Latitude>37.92000</Latitude>
  <Longitude>-109.48000</Longitude>
  <Altitude>22000</Altitude>
  <TimeOffset>50</TimeOffset>
</WayPoint>
<WayPoint>
  <Id>3</Id>
  <Latitude>39.42000</Latitude>
  <Longitude>-101.51000</Longitude>
  <Altitude>30000</Altitude>
  <TimeOffset>90</TimeOffset>
</WayPoint>
<Alarm>
  <Id>2534</Id>
  <Type>EDRTurbulence</Type>
  <Condition>
```

Flight Route Alerting and Briefing – API

```
<PropertyIsGreaterThan>
  <Name>EDR</Name>
  <Value>0.2</Value>
</PropertyIsGreaterThan>
</Condition>
</Alarm>
<Alarm>
  <Id>2535</Id>
  <Type>Sigmet</Type>
  <Condition>
    <PropertyIsEqualTo>
      <Name>Type</Name>
      <Value>Convective</Value>
    </PropertyIsEqualTo>
  </Condition>
</Alarm>
<Alarm>
  <Id>2536</Id>
  <Type>Lightning</Type>
  <Condition>
    <Exists/>
  </Condition>
</Alarm>
<Alarm>
  <Id>2537</Id>
  <Type>USTFR</Type>
  <Condition>
    <PropertyIsNotEqualTo>
      <Name>Type</Name>
      <Value>Stadium</Value>
    </PropertyIsNotEqualTo>
  </Condition>
</Alarm>
<Alarm>
  <Id>2538</Id>
  <Type>ForecastCeiling</Type>
  <Condition>
    <PropertyIsLessThan>
      <Name>Altitude</Name>
      <Value>10000</Value>
    </PropertyIsLessThan>
  </Condition>
</Alarm>
</Route>
</FlightPlan>
```

Flight Route Alerting and Briefing – API

7 Weather products Tables

The following table contains a list of all possible weather alarms that can be set for the various components of the Flight Plan as well as examples of possible values.

ROUTE					
Alarm Type	Property	Example Value	Value Type	Units / Ranges	Coverage Area
USTFR	Type	Fire	String - enumeration	Type (Fire, Nuclear, VIP, Stadium, Other)	US
EDRTurbulence	EDR	0.2	String - enumeration	EDR (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)	Global (80N-80S)
EDRConvectiveTurbulence	EDR	0.2	String - enumeration	EDR (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)	Global (80N-80S)
EDRNonConvectiveTurbulence	EDR	0.2	String - enumeration	EDR (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)	Global (80N-80S)
EDRNOWCast	EDR	0.2	String – enumeration	EDR (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)	Global (80N-80S)
FHIcing	Category	Severe	String - enumeration	Category (Slight, Moderate, Severe)	Global (80N-80S)
HighIceWaterContent	Category	Severe	String- enumeration	Category (Slight, Moderate, Severe)	Global (80N-80S)
FHThunderstorms	<i>N/A – use Exists operator</i>				Global (80N-80S)
USConvectiveSigmet	<i>N/A – use Exists operator</i>				Continental US
USIcingAirmet	<i>N/A – use Exists operator</i>				Continental US
USTurbulenceAirmet	<i>N/A – use Exists operator</i>				Continental US
USMobsAirmet	<i>N/A – use Exists operator</i>				Continental US

Flight Route Alerting and Briefing – API

Sigmet	Type	Thunderstorms	String - enumeration	Type (Turbulence, Thunderstorms, Icing, VolcanicAsh, Other – note that Convective is deprecated – use Thunderstorms instead)	Global
VolcanicAdvisory	<i>N/A – use Exists operator</i>				Global
Lightning	<i>N/A – use Exists operator</i>				Global
ForecastCeiling	Altitude	10000	Integer	Feet	Global
ForecastVisibility	Distance	10	Integer	Statute miles	Global
ObservedCeiling	Altitude	10000	Integer	Feet	Global
ObservedVisibility	Distance	10	Integer	Statute miles	Global
Radar	DBZ	10	String - enumeration	DBZ (5, 10, ..., 70, 75) Note: If omitted, any DBZ value will generate an alert for requested Type value(s)	Global
	Type	Snow	String - enumeration	Type (Rain, Mix, Snow) Note: If omitted, any Type value will generate an alert for requested DBZ value(s)	
ObservedPrecipitation	Category	Moderate	String - enumeration	Category (Light, Moderate, Heavy) Note: If omitted, any Category value will generate an alert for requested Type value(s)	Global
	Type	Snow	String - enumeration	Type (Rain, Mix, Snow) Note: If omitted, any Type value will generate an alert for requested Category value(s)	

Flight Route Alerting and Briefing – API

AIRPORT					
Alarm Type	Property	Example Value	Value Type	Units / Ranges	
ForecastCeiling	Altitude	10000	Integer	Feet	Global
ForecastVisibility	Distance	10	Integer	Statute miles	Global
ObservedCeiling	Altitude	10000	Integer	Feet	Global
ObservedVisibility	Distance	10	Integer	Statute miles	Global
Lightning	<i>N/A – use Exists operator</i>				Global
PossibleTornado	<i>N/A – use Exists operator</i>				Continental US
SevereHail	<i>N/A – use Exists operator</i>				Global
AnyHail	<i>N/A – use Exists operator</i>				Global
SurfaceWinds	WindSpeed	10	Integer	Knots	Global
	WindDirection	270	Integer	Degrees	
	WindGust	20	Integer	Knots	
ObservedFog	<i>N/A – use Exists operator</i>				Global
METAR	FlightCategory	IFR	String - enumeration	Flight Category (VFR, MVFR, IFR, LIFR)	Global
TAF	FlightCategory	IFR	String - enumeration	Flight Category (VFR, MVFR, IFR, LIFR)	Global

Flight Route Alerting and Briefing – API

Radar	DBZ	10	String - enumeration	DBZ (5, 10, ..., 70, 75) Note: If omitted, any DBZ value will generate an alert for requested Type value(s)	Global
	Type	Snow	String - enumeration	Type (Rain, Mix, Snow) Note: If omitted, any Type value will generate an alert for requested DBZ value(s)	
ObservedPrecipitation	Category	Moderate	String - enumeration	Category (Light, Moderate, Heavy) Note: If omitted, any Category value will generate an alert for requested Type value(s)	Global
	Type	Snow	String - enumeration	Type (Rain, Mix, Snow) Note: If omitted, any Type value will generate an alert for requested Category value(s)	

The following table contains the possible operators that can be used in alarm configuration for the above weather products.

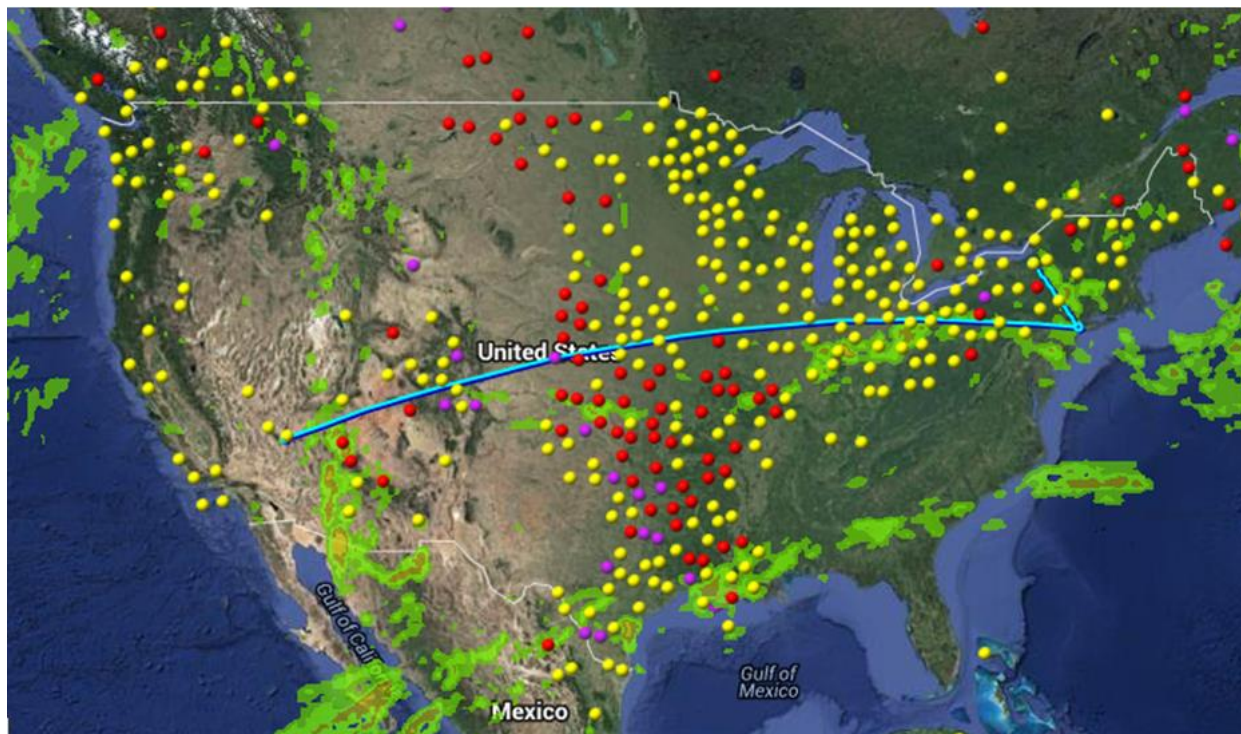
Comparison Operators
PropertyIsEqualTo
PropertyIsNotEqualTo
PropertyIsLessThan
PropertyIsLessThanOrEqualTo
PropertyIsGreaterThan
PropertyIsGreaterThanOrEqualTo
PropertyIsNull
Logical Operators
And
Or
Not
Special Operator
Exists

8 Appendix 1. – Flight Plan Alerting Use Case

Flight Plan Alerting Use Case

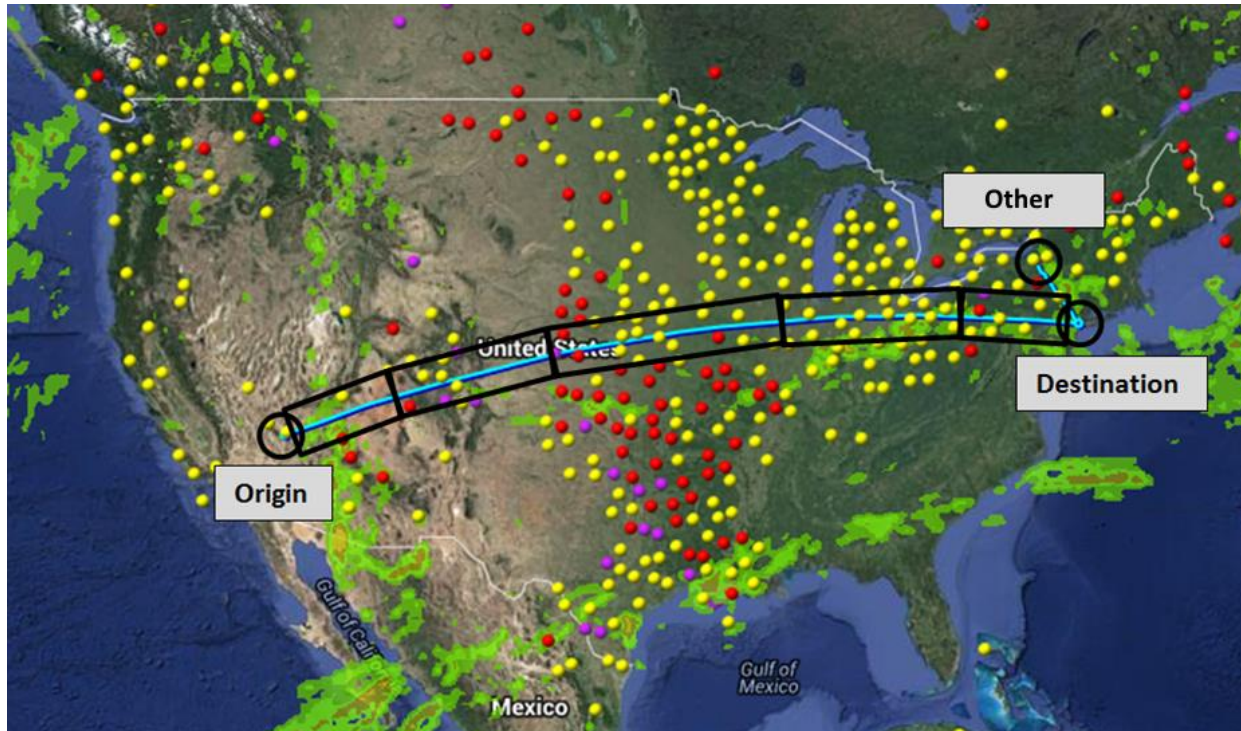
Example:

Flight from Las Vegas to New York



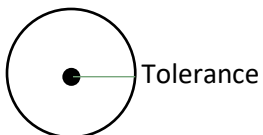
In this simple example, there are 5 flight segments in the Flight Plan, along with an origin airport, destination airport, and other airport:

Flight Route Alerting and Briefing – API



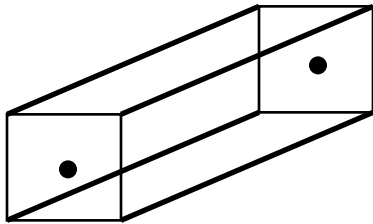
The Origin airport, Destination airport, and Other airport are represented by a dot with a radius. Each airport has an ID, ICAO ID, latitude, longitude, type (Origin, Destination, or Other) and tolerance for the radius. An Other airport has optionally a start valid time and an end valid time. These values can be used so that alerts will only be generated within the valid time range.

Each flight plan must have one Origin airport and one Destination airport. A flight plan may have up to 25 Other airports – these are optional.

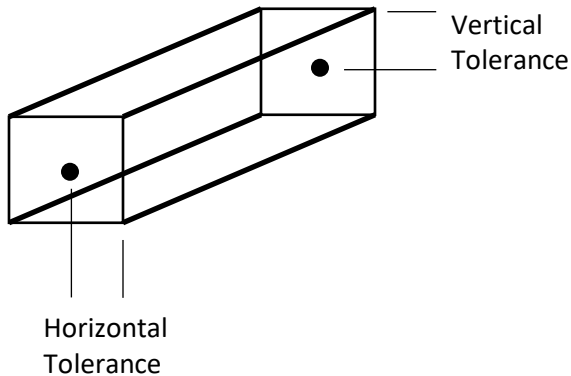


Flight Route Alerting and Briefing – API

The waypoints combine to constitute flight segments. A waypoint consists of an ID, latitude, longitude, altitude, and time offset (in minutes since estimated departure time). Each flight segment can be represented as a three dimensional rectangle. The first dot is the first waypoint for the flight segment, and the second dot is the second waypoint.



Flight segments are normalized to be the same altitude from start to finish based on averaging the starting and ending altitude.



It is optional to define Climb and Descent WayPoints. If defined, there must be a minimum of two Climb WayPoints and a minimum of two Descent WayPoints. If a WayPoint is set to type Climb, then the Tolerance Climb Vertical and Tolerance Climb Horizontal values are used instead of the Tolerance Cruise Vertical and Tolerance Cruise Horizontal values. If a WayPoint is set to type Descent, then the Tolerance Descent Vertical and Tolerance Descent Horizontal values are used instead of the Tolerance Cruise Vertical and Tolerance Cruise Horizontal values. If a WayPoint is set to type Cruise or not set at all, then the Tolerance Cruise Vertical and Tolerance Cruise Horizontal values are used.

In this example, the width of the rectangle is 100 nautical miles, the height is 2000 feet, and the segments are numbered 1 – 5.

Flight Route Alerting and Briefing – API

Origin Airport

- Airport: KLAS
 - Lat: 36.08
 - Long: -115.15

Destination Airport

- Airport: KJFK
 - Lat: 40.64
 - Long: -73.78

Other Airport

- Airport: KSYR
 - Lat: 43.11
 - Long: -76.11

Segment 1

- | | |
|--|--|
| <ul style="list-style-type: none">• Waypoint 1<ul style="list-style-type: none">○ Lat: 36.53○ Long: -114.61○ Altitude: 16000 ft○ Time: 20 minutes | <ul style="list-style-type: none">• Waypoint 2<ul style="list-style-type: none">○ Lat: 37.92○ Long: -109.48○ Altitude: 22000 ft○ Time: 50 minutes |
|--|--|

Segment 2

- | | |
|--|---|
| <ul style="list-style-type: none">• Waypoint 2<ul style="list-style-type: none">○ Lat: 37.92○ Long: -109.48○ Altitude: 22000 ft○ Time: 50 minutes | <ul style="list-style-type: none">• Waypoint 3<ul style="list-style-type: none">○ Lat: 39.42○ Long: -101.51○ Altitude: 30000 ft○ Time: 120 minutes |
|--|---|

Segment 3

Flight Route Alerting and Briefing – API

- Waypoint 3
 - Lat: 39.42
 - Long: -101.51
 - Altitude: 30000 ft
 - Time: 120 minutes

- Waypoint 4
 - Lat: 41.16
 - Long: -89.73
 - Altitude: 30000 ft
 - Time: 240 minutes

Segment 4

- Waypoint 4
 - Lat: 41.16
 - Long: -89.73
 - Altitude: 34000 ft
 - Time: 240 minutes

- Waypoint 5
 - Lat: 41.03
 - Long: -80.94
 - Altitude: 22000 ft
 - Time: 340 minutes

Segment 5

- Waypoint 5
 - Lat: 41.03
 - Long: -80.94
 - Altitude: 22000 ft
 - Time: 340 minutes

- Waypoint 6
 - Lat: 40.91
 - Long: -74.62
 - Altitude: 16000 ft
 - Time: 400 minutes

To construct the XML for a flight plan, consider the following simplified snippet of XML:

XML

```
<FlightPlan>
  <Airport>
  </Airport>
  <Airport>
  </Airport>
  <Airport>
  </Airport>
  <Route>
    <WayPoint>
    </WayPoint>
    <WayPoint>
    </WayPoint>
    <WayPoint>
```

Flight Route Alerting and Briefing – API

```
</WayPoint>
<WayPoint>
</WayPoint>
<WayPoint>
</WayPoint>
<WayPoint>
</WayPoint>
</Route>
</FlightPlan>
```

The first components are Airports – an Origin Airport, a Destination Airport, and up to 25 Other Airports. Each of these airports can have a set of Alarms set up on an individual basis.

The next main component of the Flight Plan is the Route. The Route consists of a list of WayPoints, but the Route itself has one set of Alarms for the entire Route.

Now the simple example of XML is expanded to match the flight from Las Vegas to New York:

```
XML
<FlightPlan
  <Airport>
    <Id>KLAS</Id>
    <ICAOId>KLAS</ICAOId>
    <Type>Origin</Type>
    <Tolerance>50</Tolerance>
    <Latitude>36.080056</Latitude>
    <Longitude>-115.15225</Longitude>
  </Airport>
  <Airport>
    <Id>KJFK</Id>
    <ICAOId>KJFK</ICAOId>
    <Type>Destination</Type>
    <Tolerance>50</Tolerance>
    <Latitude>40.639751</Latitude>
    <Longitude>-73.778926</Longitude>
  </Airport>
  <Airport>
    <Id>KSYR</Id>
    <ICAOId>KSYR</ICAOId>
    <Type>Other</Type>
    <Tolerance>50</Tolerance>
    <Latitude>43.111194</Latitude>
    <Longitude>-76.106306</Longitude>
```

```
</Airport>
<Route>
  <VerticalTolerance>1000</VerticalTolerance>
  <HorizontalTolerance>50</HorizontalTolerance>
  <WayPoint>
    <Id>1</Id>
    <Latitude>36.53</Latitude>
    <Longitude>-114.61</Longitude>
    <Altitude>16000</Altitude>
    <TimeOffset>20</TimeOffset>
  </WayPoint>
  <WayPoint>
    <Id>2</Id>
    <Latitude>37.92</Latitude>
    <Longitude>-109.48</Longitude>
    <Altitude>22000</Altitude>
    <TimeOffset>50</TimeOffset>
  </WayPoint>
  <WayPoint>
    <Id>3</Id>
    <Latitude>39.42</Latitude>
    <Longitude>-101.51</Longitude>
    <Altitude>30000</Altitude>
    <TimeOffset>30</TimeOffset>
  </WayPoint>
  <WayPoint>
    <Id>4</Id>
    <Latitude>41.16</Latitude>
    <Longitude>-89.731</Longitude>
    <Altitude>30000</Altitude>
    <TimeOffset>240</TimeOffset>
  </WayPoint>
  <WayPoint>
    <Id>5</Id>
    <Latitude>41.03</Latitude>
    <Longitude>-80.94</Longitude>
    <Altitude>22000</Altitude>
    <TimeOffset>340</TimeOffset>
  </WayPoint>
  <WayPoint>
    <Id>6</Id>
    <Latitude>40.91</Latitude>
    <Longitude>-74.62</Longitude>
    <Altitude>16000</Altitude>
    <TimeOffset>400</TimeOffset>
  </WayPoint>
</Route>
</FlightPlan>
```

Next, Alarms are added to the XML. An Alarm is defined by an ID, type, and a condition. Consider the following simplified snippet of XML:

XML
<pre><Alarm> <Id></Id> <Type></Type> <Condition> </Condition> </Alarm></pre>

A condition is made up of one to many operators. At the outermost level can be one of the logical operators, And, Or or Not. These are optional, and can be used to combine operators. Logical operators cannot be nested.

The comparison operators are for comparing a property's value against an input value. For example, for the Alarm type EDRTurbulence, a comparison operator would be PropertyIsGreaterThan.

The special operator Exists is used to check for the mere existence of the Alarm type. An example for using this operator is the Alarm type Lightning.

Now the XML snippet is expanded to add in example Alarms:

XML
<pre><Alarm> <Id>1</Id> <Type>Lightning</Type> <Condition> <Exists/> </Condition> </Alarm> <Alarm> <Id>2</Id> <Type>EDRTurbulence</Type> <Condition> <PropertyIsGreaterThan> <Name>EDR</Name> <Value>0.3</Value> </PropertyIsGreaterThan> </Condition></pre>

Flight Route Alerting and Briefing – API

```
</Alarm>
```

The flight plan itself requires certain information, such as ID, flight date, tail number, flight number, aircraft type, scheduled departure time, and estimated departure time.

Below is the complete example of XML for the flight from Las Vegas to New York, with the flight plan information and alarms added in.

```
XML
<FlightPlan xmlns="http://weather-services.telventdn.com/alerting/flightplan/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://cat-weather-
alerting.dtn.com/xsds/alerting/flightplan/flightPlanAlert.xsd">
  <Name>Las Vegas to New York</Name>
  <FlightDate>2015-01-14</FlightDate>
  <TailNumber>567</TailNumber>
  <FlightNumber>244</FlightNumber>
  <AircraftType>B777</AircraftType>
  <ScheduledDepartureTime>2015-01-14T15:00:00Z</ScheduledDepartureTime>
  <EstimatedDepartureTime>2015-01-14T15:00:00Z</EstimatedDepartureTime>
  <EstimatedArrivalTime>2015-01-14T22:00:00Z</EstimatedArrivalTime>
  <Airport>
    <Id>KLAS</Id>
    <ICAOId>KLAS</ICAOId>
    <Type>Origin</Type>
    <Tolerance>50</Tolerance>
    <Latitude>36.080056</Latitude>
    <Longitude>-115.15225</Longitude>
    <Alarm>
      <Id>1</Id>
      <Type>SurfaceWinds</Type>
      <Condition>
        <PropertyIsGreaterThan>
          <Name>WindSpeed</Name>
          <Value>15</Value>
        </PropertyIsGreaterThan>
      </Condition>
    </Alarm>
    <Alarm>
      <Id>2</Id>
      <Type>Lightning</Type>
      <Condition>
        <Exists/>
      </Condition>
    </Alarm>
  </Airport>
</FlightPlan>
```



```
<Id>KJFK</Id>
<ICAOId>KJFK</ICAOId>
<Type>Destination</Type>
<Tolerance>50</Tolerance>
<Latitude>40.639751</Latitude>
<Longitude>-73.778926</Longitude>
<Alarm>
  <Id>3</Id>
  <Type>SurfaceWinds</Type>
  <Condition>
    <PropertyIsGreaterThan>
      <Name>WindSpeed</Name>
      <Value>15</Value>
    </PropertyIsGreaterThan>
  </Condition>
</Alarm>
<Alarm>
  <Id>4</Id>
  <Type>Lightning</Type>
  <Condition>
    <Exists/>
  </Condition>
</Alarm>
</Airport>
<Airport>
  <Id>KSYR</Id>
  <ICAOId>KSYR</ICAOId>
  <Type>Other</Type>
  <Tolerance>50</Tolerance>
  <Latitude>43.111194</Latitude>
  <Longitude>-76.106306</Longitude>
  <Alarm>
    <Id>5</Id>
    <Type>SurfaceWinds</Type>
    <Condition>
      <PropertyIsGreaterThan>
        <Name>WindSpeed</Name>
        <Value>15</Value>
      </PropertyIsGreaterThan>
    </Condition>
  </Alarm>
  <Alarm>
    <Id>6</Id>
    <Type>Lightning</Type>
    <Condition>
      <Exists/>
    </Condition>
  </Alarm>
</Airport>
```

```
</Airport>
<Route>
  <VerticalTolerance>1000</VerticalTolerance>
  <HorizontalTolerance>50</HorizontalTolerance>
  <WayPoint>
    <Id>1</Id>
    <Latitude>36.53</Latitude>
    <Longitude>-114.61</Longitude>
    <Altitude>16000</Altitude>
    <TimeOffset>20</TimeOffset>
  </WayPoint>
  <WayPoint>
    <Id>2</Id>
    <Latitude>37.92</Latitude>
    <Longitude>-109.48</Longitude>
    <Altitude>22000</Altitude>
    <TimeOffset>50</TimeOffset>
  </WayPoint>
  <WayPoint>
    <Id>3</Id>
    <Latitude>39.42</Latitude>
    <Longitude>-101.51</Longitude>
    <Altitude>30000</Altitude>
    <TimeOffset>80</TimeOffset>
  </WayPoint>
  <WayPoint>
    <Id>4</Id>
    <Latitude>41.16</Latitude>
    <Longitude>-89.731</Longitude>
    <Altitude>30000</Altitude>
    <TimeOffset>240</TimeOffset>
  </WayPoint>
  <WayPoint>
    <Id>5</Id>
    <Latitude>41.03</Latitude>
    <Longitude>-80.94</Longitude>
    <Altitude>22000</Altitude>
    <TimeOffset>340</TimeOffset>
  </WayPoint>
  <WayPoint>
    <Id>6</Id>
    <Latitude>40.91</Latitude>
    <Longitude>-74.62</Longitude>
    <Altitude>16000</Altitude>
    <TimeOffset>400</TimeOffset>
  </WayPoint>
  <Alarm>
    <Id>7</Id>
```

Flight Route Alerting and Briefing – API

```
<Type>EDRTurbulence</Type>
<Condition>
  <PropertyIsGreaterThan>
    <Name>EDR</Name>
    <Value>0.3</Value>
  </PropertyIsGreaterThan>
</Condition>
</Alarm>
<Alarm>
  <Id>8</Id>
  <Type>Lightning</Type>
  <Condition>
    <Exists/>
  </Condition>
</Alarm>
</Route>
</FlightPlan>
```

9 Appendix 2. – Error Codes

Error Codes

```
<Error>
  <Code>FP-50</Code>
  <Type>BusinessLogicError</Type>
  <Description>Duplicate Airport ICAO id found: SCEL </Description>
</Error>
```

```
<Error>
  <Code>FP-51</Code>
  <Type>BusinessLogicError</Type>
  <Description>Duplicate Alarm id found: 2,539 </Description>
</Error>
```

```
<Error>
  <Code>FP-52</Code>
  <Type>BusinessLogicError</Type>
  <Description>Duplicate Way Point id found: RESNO</Description>
</Error>
```

```
<Error>
  <Code>FP-54</Code>
  <Type>BusinessLogicError</Type>
  <Description>No flight plan found for: Flight Plan Id: 1430922449211, Flight Date: 2015-05-06,
Scheduled Departure Time: 2015-05-06T15:27:00Z</Description>
</Error>
```

```
<Error>
  <Code>FP-56</Code>
  <Type>BusinessLogicError</Type>
  <Description>Flight plan already exists: Flight Plan Id: 51345, Flight Date: 2015-01-26, Scheduled
Departure Time: 2015-01-26T15:00:00Z</Description>
</Error>
```

Flight Route Alerting and Briefing – API

```
<Error>
  <Code>FP-58</Code>
  <Type>BusinessLogicError</Type>
  <Description>Flight plan already being tracked: Flight Plan Id: 1429132547748, Flight Date: 2015-04-15, Scheduled Departure Time: 2015-04-15T22:15:00Z  </Description>
</Error>
```

```
<Error>
  <Code>FP-59</Code>
  <Type>BusinessLogicError</Type>
  <Description>No tracked Flight plan found for: Flight Plan Id: 51345, Flight Date: 2015-01-26, Scheduled Departure Time: 2015-01-26T15:00:00Z  </Description>
</Error>
```

```
  <Code>FP-60</Code>
  <Type>ValidationError</Type>
  <Description>Validation Error: Error at line: 3 and Column: 9, cvc-complex-type.2.4.a: Invalid content was found starting with element 'Name'. One of '{"http://weather-alerting.dtn.com/flightplan/v1":Id}' is expected.</Description>
</Error>
```

```
<Error>
  <Code>FP-61</Code>
  <Type>BusinessLogicError</Type>
  <Description>Cannot start tracking a plan whose arrival time is in the past: Flight Plan Id: 51345, Flight Date: 2015-01-26, Scheduled Departure Time: 2015-01-26T15:00:00Z, Estimated Arrival Time: 2015-01-26T22:30:00Z</Description>
</Error>
```

```
<Error>
  <Code>FP-62</Code>
  <Type>BusinessLogicError</Type>
  <Description>Estimated arrival time cannot be before scheduled departure time: Flight Plan Id: 51345, Flight Date: 2015-01-26, Scheduled Departure Time: 2015-01-26T15:00:00Z, Estimated Arrival Time: 2015-01-24T22:30:00Z  </Description>
</Error>
```

Flight Route Alerting and Briefing – API

```
<Error>
  <Code>FP-63</Code>
  <Type>BusinessLogicError</Type>
  <Description>Incorrect ICAO format: KTEB1.</Description>
</Error>
```

```
<Error>
  <Code>FP-64</Code>
  <Type>BusinessLogicError</Type>
  <Description>Need at least one airport or two waypoints to process the flight plan.</Description>
</Error>
```

```
<Error>
  <Code>FP-65</Code>
  <Type>ValidationError</Type>
  <Description>Validation Error: Property Length not supported for alarm type
ForecastVisibility</Description>
</Error>
```

```
<Error>
  <Code>FP-66</Code>
  <Type>BusinessLogicError</Type>
  <Description>Cannot start tracking a deactivated flight plan: Flight Plan Id: 0713_1, Flight Date:
2015-01-14, Scheduled Departure Time: 2015-04-08T20:00:00Z   </Description>
</Error>
```

```
<Error>
  <Code>FP-68</Code>
  <Type>BusinessLogicError</Type>
  <Description>Specify either none or both (StartValidTime and EndValidTime): KTEB.</Description>
</Error>
```

```
<Error>
  <Code>FP-70</Code>
  <Type>SystemError</Type>
  <Description>An unknown problem has occurred. Please try again later or contact support at
AviationSupport@TelventDTN.com. </Description>
</Error>
```

<pre><Error> <Code>FP-71</Code> <Type>BusinessLogicError</Type> <Description>Unsupported REQUESTTYPE: EvaluateNow. </Description> </Error></pre>
<pre><Error> <Code>FP-72</Code> <Type>BusinessLogicError</Type> <Description>Incorrect type for maxAge (only integer values accepted): 36f.</Description> </Error></pre>
<pre><Error> <Code>FP-73</Code> <Type>BusinessLogicError</Type> <Description>A minimum of two descent way points are required, if present.</Description> </Error></pre>
<pre><Error> <Code>FP-73</Code> <Type>BusinessLogicError</Type> <Description>A minimum of two climb way points are required, if present. </Description> </Error></pre>
<pre><Error> <Code>FP-74</Code> <Type>BusinessLogicError</Type> <Description>WayPoint Id DYP SO is out of order. Climb and Descent waypoints should be in consecutive blocks.</Description> </Error></pre>
<pre><Error> <Code>FP-75</Code> <Type>BusinessLogicError</Type> <Description>Longitude out of range for waypoint: AIR.</Description> </Error></pre>
<pre><Error> <Code>FP-77</Code></pre>

Flight Route Alerting and Briefing – API

```
<Type>BusinessLogicError</Type>
<Description>Incorrect type for computeHasAlerts (only true/false accepted): trued.</Description>
</Error>

<Error>
  <Code>FP-78</Code>
  <Type>BusinessLogicError</Type>
  <Description>Incorrect type for activeOnly (only true/false accepted): trued.</Description>
</Error>

<Error>
  <Code>FP-79</Code>
  <Type>BusinessLogicError</Type>
  <Description>At least one of these update parameters is required: EstimatedDepartureTime or
EstimatedArrivalTime.</Description>
</Error>

<Error>
  <Code>FP-80</Code>
  <Type>BusinessLogicError</Type>
  <Description>Cannot update a deactivated flight plan: Flight Plan Id: 0713_1, Flight Date: 2015-01-
14, Scheduled Departure Time: 2015-04-08T20:00:00Z .</Description>
</Error>

<Error>
  <Code>FP-81</Code>
  <Type>BusinessLogicError</Type>
  <Description>Attempted to update with an identical flight plan. Flight Plan Id: 0713_1, Flight Date:
2015-01-14, Scheduled Departure Time: 2015-04-08T20:00:00Z .</Description>
</Error>
```

10 Appendix 3. – Info CODES

Info Codes

```
<Info>
  <Code>FP-20</Code>
  <Description>There is no METAR data coverage available for the requested airport: KTEB.</Description>
```


Flight Route Alerting and Briefing – API

</Info>

<Info>

<Code>FP-21</Code>

<Description>There is no TAF data coverage available for the requested airport: KTEB.</Description>

</Info>

<Info>

<Code>FP-22</Code>

<Description>There is no Visibility data coverage available for the requested airport:
KTEB.</Description>

</Info>

<Info>

<Code>FP-23</Code>

<Description>There is no Ceiling data coverage available for the requested airport: KTEB.</Description>

</Info>

<Info>

<Code>FP-24</Code>

<Description>There are segments of the requested route that fall outside Visibility data coverage areas.
Alerts will be limited..</Description>

</Info>

<Info>

<Code>FP-25</Code>

<Description> There are segments of the requested route that fall outside Ceiling data coverage areas.
Alerts will be limited.</Description>

</Info>

<Info>

<Code>FP-26</Code>

<Description>There is no Radar data coverage available for the requested airport: KTEB.</Description>

</Info>

<Info>

Flight Route Alerting and Briefing – API

```
<Code>FP-27</Code>  
<Description> There are segments of the requested route that fall outside Radar data coverage areas.  
Alerts will be limited.</Description>  
</Info>
```

11 Appendix 4. – Demonstration Application

Demonstration Application

A demonstration application is available here:

<https://cat-weather-alerting.dtn.com/tools/demo/FlightRouteAlertingDemo.html>

Instructions:

Log in with your credentials.

Click button labeled “Load Flight Plan KORD EGLL”.

Click button labeled “Evaluate”.

Note the responses in the text fields labeled “Response” and “WebSocket Response”.

Click button labeled “Create”.

Note the response in the text field labeled “Response”.

Click button labeled “Start Tracking”.

Note the responses in the text fields labeled “Response” and “WebSocket Response”.

12 Appendix 5. – Business Rules

Business Rules

Flight Route Alerting and Briefing – API

In a FlightPlan, an Airport is limited to a maximum of 15 Alarms and a Route is limited to a maximum of 20 Alarms.

In a FlightPlan, only 1 Origin Airport and 1 Destination Airport are allowed. The maximum number of Other Airports is 25. The Origin Airport and Destination Airport may be the same airport (the same ICAO id).

In a FlightPlan, the maximum number of rules is 1000. Rules are defined as the multiplier of waypoints and alarms. For example, 111 waypoints and 9 alarms equals 999 rules.

In a FlightPlan, if the aircraft type is not specified and FHLing is one of the Alarm types, a notification is sent as part of the FlightPlanAlert stating that FRA is using “Medium” as aircraft size for a default value.

If the airport’s ICAOID is not a METAR reporting station (or if it’s not reporting currently), an alternate ICAOID within the radius specified in the flight plan is used to check for Alerts. If any METAR within the radius specified exceeds the user threshold, an alert will be returned with the ID of the exceeding station.

If the airport’s ICAOID is not a TAF reporting station (or if it’s not reporting currently), an alternate ICAOID within the radius specified in the flight plan is used to check for Alerts. If any TAF within the radius specified exceeds the user threshold, an Alert will be returned with the ID of the exceeding station.

FlightPlan Alarms for Route segments do not conform to great circle route. They are direct point to point lines. If more precise alerting is required, the user must supply extra WayPoints.

13 Appendix 6. – Implementation Notes

It is important in using WebSockets to have retry / reconnect logic, as a WebSocket is intended to always be connected. In certain circumstances a WebSocket may disconnect, so handling that case is supported in the API. See websocket.org for more information.

The WebSocket client software must support partial messages. Depending on the client implementation, support for this can be turned on explicitly. For example, in Java, the class `MessageHandler.Partial` (`javax.websocket`) is used to support partial messages. See <http://docs.oracle.com/javaee/7/api/javax/websocket/MessageHandler.Partial.html> for more information.

14 Appendix 7. – Alert Responses

Alarm Type	
USTFR	<pre> <Alert> <AlarmId>2541</AlarmId> <AlarmType>USTFR</AlarmType> <Time>2016-02-03T03:31:38.962Z</Time> <Data> <Property> <Name>ValidTime</Name> <Value>2015-09-16T12:05:14.00Z</Value> </Property> <Property> <Name>ExpirationTime</Name> <Value>2070-01-01T00:00:00.00Z</Value> </Property> </Data> </Alert> </pre>
EDRTurbulence	<pre> <Alert> <AlarmId>2538</AlarmId> <AlarmType>EDRTurbulence</AlarmType> <Time>2016-02-03T03:03:30.691Z</Time> <Data> <Property> <Name>EDR</Name> <Value>0.4</Value> </Property> <Property> <Name>ValidTime</Name> <Value>2016-02-03T12:00:00.00Z</Value> </Property> <Property> <Name>FlightLevel</Name> <Value>FL390</Value> </Property> </Data> </Alert> </pre>
FHIcing	<pre> <Alert> </pre>

Flight Route Alerting and Briefing – API

	<pre><AlarmId>2547</AlarmId> <AlarmType>FHIcing</AlarmType> <Time>2016-02-04T01:16:32.491Z</Time> <Data> <Property> <Name>ValidTime</Name> <Value>2016-02-04T03:00:00.00Z</Value> </Property> <Property> <Name>FlightLevel</Name> <Value>FL270</Value> </Property> </Data> </Alert></pre>
FHThunderstorms	<pre><Alert> <AlarmId>2539</AlarmId> <AlarmType>FHThunderstorms</AlarmType> <Time>2016-02-03T03:11:03.288Z</Time> <Data> <Property> <Name>ValidTime</Name> <Value>2016-02-03T06:00:00.00Z</Value> </Property> <Property> <Name>Top</Name> <Value>37000</Value> </Property> </Data> </Alert></pre>
USConvectiveSigmet	<pre><Alert> <AlarmId>2542</AlarmId> <AlarmType>USConvectiveSigmet</AlarmType> <Time>2016-02-03T03:31:38.961Z</Time> <Data> <Property> <Name>ExpirationTime</Name> <Value>2016-02-03T04:55:00.00Z</Value> </Property></pre>

Flight Route Alerting and Briefing – API

	<pre><Property> <Name>Top</Name> <Value>39000</Value> </Property> <Property> <Name>ValidTime</Name> <Value>2016-02-03T02:55:00.00Z</Value> </Property> </Data> </Alert></pre>
USIcingAirmet	<pre><Alert> <AlarmId>2543</AlarmId> <AlarmType>USIcingAirmet</AlarmType> <Time>2016-02-03T03:38:31.405Z</Time> <Data> <Property> <Name>ExpirationTime</Name> <Value>2016-02-03T09:00:00.00Z</Value> </Property> <Property> <Name>Bottom</Name> <Value>0</Value> </Property> <Property> <Name>ValidTime</Name> <Value>2016-02-03T02:45:00.00Z</Value> </Property> <Property> <Name>Top</Name> <Value>65000</Value> </Property> </Data> </Alert></pre>
USTurbulenceAirmet	<pre><Alert> <AlarmId>2544</AlarmId> <AlarmType>USTurbulenceAirmet</AlarmType> <Time>2016-02-03T03:31:38.961Z</Time> <Data></pre>

Flight Route Alerting and Briefing – API

	<pre><Property> <Name>ExpirationTime</Name> <Value>2016-02-03T09:00:00.00Z</Value> </Property> <Property> <Name>Bottom</Name> <Value>18000</Value> </Property> <Property> <Name>ValidTime</Name> <Value>2016-02-03T02:45:00.00Z</Value> </Property> <Property> <Name>Top</Name> <Value>39000</Value> </Property> </Data> </Alert></pre>
USMobsAirmet	<pre><Alert> <AlarmId>2545</AlarmId> <AlarmType>USMobsAirmet</AlarmType> <Time>2016-02-03T03:31:38.971Z</Time> <Data> <Property> <Name>ExpirationTime</Name> <Value>2016-02-03T09:00:00.00Z</Value> </Property> <Property> <Name>ValidTime</Name> <Value>2016-02-03T02:45:00.00Z</Value> </Property> </Data> </Alert></pre>
Sigmet	<pre><Alert> <AlarmId>2548</AlarmId> <AlarmType>Sigmet</AlarmType> <Time>2016-02-04T01:19:39.839Z</Time> <Data></pre>

Flight Route Alerting and Briefing – API

	<pre> <Property> <Name>ExpirationTime</Name> <Value>2016-02-04T03:35:00.00Z</Value> </Property> <Property> <Name>ValidTime</Name> <Value>2016-02-03T23:35:00.00Z</Value> </Property> </Data> </Alert> </pre>
VolcanicAdvisory	<pre> <Alert> <AlarmId>2546</AlarmId> <AlarmType>VolcanicAdvisory</AlarmType> <Time>2016-02-03T03:38:31.408Z</Time> <Data> <Property> <Name>Bottom</Name> <Value>0</Value> </Property> <Property> <Name>ValidTime</Name> <Value>2016-02-02T19:00:00.00Z</Value> </Property> <Property> <Name>Top</Name> <Value>65000</Value> </Property> </Data> </Alert> </pre>
Lightning	<pre> <Alert> <AlarmId>2537</AlarmId> <AlarmType>Lightning</AlarmType> <Time>2016-02-03T03:38:31.405Z</Time> <Data> <Property> <Name>ValidTime</Name> <Value>2016-02-03T03:36:00.00Z</Value> </Property> </pre>

Flight Route Alerting and Briefing – API

	<pre></Data> </Alert></pre>
ForecastCeiling	<pre><Alert> <AlarmId>4501</AlarmId> <AlarmType>ForecastCeiling</AlarmType> <Time>2016-02-03T03:25:37.372Z</Time> <Data> <Property> <Name>ICAOId</Name> <Value>KORD</Value> </Property> <Property> <Name>ForecastValidTime</Name> <Value>2016-02-03T01:00:00.00Z</Value> </Property> <Property> <Name>CeilingCode</Name> <Value>OVC</Value> </Property> <Property> <Name>Altitude</Name> <Value>400</Value> </Property> <Property> <Name>ForecastExpirationDateTime</Name> <Value>2016-02-03T03:00:00.00Z</Value> </Property> </Data> </Alert></pre>
ForecastVisibility	<pre><Alert> <AlarmId>4500</AlarmId> <AlarmType>ForecastVisibility</AlarmType> <Time>2016-02-03T03:21:02.267Z</Time> <Data> <Property> <Name>ICAOId</Name> <Value>KORD</Value> </Property></pre>

Flight Route Alerting and Briefing – API

	<pre> <Property> <Name>ForecastValidTime</Name> <Value>2016-02-03T01:00:00.00Z</Value> </Property> <Property> <Name>Distance</Name> <Value>1.49</Value> </Property> <Property> <Name>ForecastExpirationDateTime</Name> <Value>2016-02-03T03:00:00.00Z</Value> </Property> </Data> </Alert> </pre>
ObservedCeiling	<pre> <Alert> <AlarmId>4503</AlarmId> <AlarmType>ObservedCeiling</AlarmType> <Time>2016-02-04T02:50:22.967Z</Time> <Data> <Property> <Name>ValidTime</Name> <Value>2016-02-04T01:51:00.00Z</Value> </Property> <Property> <Name>ICAOId</Name> <Value>KORD</Value> </Property> <Property> <Name>Altitude</Name> <Value>2200</Value> </Property> </Data> </Alert> </pre>
ObservedVisibility	<pre> <Alert> <AlarmId>4502</AlarmId> <AlarmType>ObservedVisibility</AlarmType> <Time>2016-02-04T02:50:22.966Z</Time> <Data> </pre>

Flight Route Alerting and Briefing – API

	<pre> <Property> <Name>ValidTime</Name> <Value>2016-02-04T01:51:00.00Z</Value> </Property> <Property> <Name>ICAOId</Name> <Value>KORD</Value> </Property> <Property> <Name>Distance</Name> <Value>9.000000</Value> </Property> </Data> </Alert> </pre>
PossibleTornado	
SevereHail	
AnyHail	
SurfaceWinds	
ObservedFog	
METAR	<pre> <Alert> <AlarmId>2527</AlarmId> <AlarmType>METAR</AlarmType> <Time>2016-02-03T03:11:03.288Z</Time> <Data> <Property> <Name>ValidTime</Name> <Value>2016-02-03T03:00:00.00Z</Value> </Property> <Property> <Name>ICAOId</Name> <Value>KHW0</Value> </Property> <Property> <Name>FlightCategory</Name> <Value>MVFR</Value> </Property> </Data> </pre>

Flight Route Alerting and Briefing – API

TAF	<pre></Alert> <Alert> <AlarmId>2532</AlarmId> <AlarmType>TAF</AlarmType> <Time>2016-02-03T03:03:30.691Z</Time> <Data> <Property> <Name>ICAOId</Name> <Value>EGMC</Value> </Property> <Property> <Name>ForecastValidTime</Name> <Value>2016-02-03T09:00:00.00Z</Value> </Property> <Property> <Name>FlightCategory</Name> <Value>MVFR</Value> </Property> <Property> <Name>ForecastExpirationDateTime</Name> <Value>2016-02-03T11:00:00.00Z</Value> </Property> </Data> </Alert></pre>
Radar	<pre><Alert> <AlarmId>2532</AlarmId> <AlarmType>Radar</AlarmType> <Time>2016-02-03T03:03:30.691Z</Time> <Data> <Property> <Name>ValidTime</Name> <Value>2016-02-03T09:00:00.00Z</Value> </Property> <Property> <Name>DBZ</Name> <Value>15</Value> </Property> <Property></pre>

Flight Route Alerting and Briefing – API

```
<Name>Type</Name>
<Value>Rain</Value>
</Property>
</Data>
</Alert>
```

15 Appendix 8: EFF XML Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EFUSUB xmlns="http://aeec.aviation-ia.net/633">
  <M633Header versionNumber="3" timestamp="2017-01-31T16:02:11.141Z"/>
  <M633SupplementaryHeader>
    <Flight flightOriginDate="2017-02-06Z" scheduledTimeOfDeparture="2017-02-06T11:10:00Z">
      <FlightIdentification>
        <FlightNumber airlineIATACode="H0" number="218">
          <CommercialFlightNumber>H0218</CommercialFlightNumber>
        </FlightNumber>
      </FlightIdentification>
      <DepartureAirport>
        <AirportICAOCODE>KSGR</AirportICAOCODE>
      </DepartureAirport>
      <ArrivalAirport>
        <AirportICAOCODE>KAGC</AirportICAOCODE>
      </ArrivalAirport>
    </Flight>
    <Aircraft aircraftRegistration="N710DL">
      <AircraftModel>
        <AircraftICAOTYPE>CL30</AircraftICAOTYPE>
      </AircraftModel>
    </Aircraft>
  </M633SupplementaryHeader>
  <SubFolder title="_Root_Folder_">
    <SubFolder title="Airport Weather">
      <Document title="KSGR - Sugar Land Regional Airport, United States" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-f15dc50b-b06d-43a9-8ec3-f77731fad5a4" type="image/png" file="KSGR_-_Sugar_Land_Regional_Airport,_United_States.png">
        <Topic name="Weather/Airport"/>
      </Document>
      <Document title="KAGC - Allegheny County Airport, United States" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-56aaf512-f5a8-47d8-b866-5693f5e8f1cd" type="image/png" file="KAGC_-_Allegheny_County_Airport,_United_States.png">
        <Topic name="Weather/Airport"/>
      </Document>
    </SubFolder>
  </SubFolder>
</EFUSUB>
```

Flight Route Alerting and Briefing – API

```
</Document>
<Document title="KHOU - William P Hobby, United States" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-9e311551-572a-4915-a88b-075d3c2a04fc" type="image/png" file="KHOU_-_William_P_Hobby,_United_States.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KPIT - Pittsburgh Intl, United States" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z" authorProfile="dispatcher"
id="t-454a9854-91c0-4192-b8b6-563545d14234" type="image/png" file="KPIT_-_Pittsburgh_Intl,_United_States.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="AirportWeather" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z" authorProfile="dispatcher" id="t-0dd8a061-f162-
4649-9593-8e6aa21fa8e9" type="text/xml" file="AirportWeather.xml">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KSGR - Sugar Land Regional Airport, United States Ceiling Height" application="DTN" updateDateTime="2017-01-
31T16:01:56.744Z" authorProfile="dispatcher" id="t-0706c025-6db6-4738-be31-d0eef47d3c15" type="image/png" file="KSGR_-
_Sugar_Land_Regional_Airport,_United_States_Ceiling_Height.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KSGR - Sugar Land Regional Airport, United States Visibility" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-000fd302-ba5f-438f-9bf5-143d072d26b4" type="image/png" file="KSGR_-
_Sugar_Land_Regional_Airport,_United_States_Visibility.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KHOU - William P Hobby, United States Ceiling Height" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-01dbbbf7-df93-4af7-9f8f-7dbafd8fdc90" type="image/png" file="KHOU_-
_William_P_Hobby,_United_States_Ceiling_Height.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KHOU - William P Hobby, United States Visibility" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-507d6fa4-ccb2-49f8-9bc6-5bb9cbc3866d" type="image/png" file="KHOU_-_William_P_Hobby,_United_States_Visibility.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KAGC - Allegheny County Airport, United States Ceiling Height" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-236790fa-0ab6-4b12-b278-037405250b41" type="image/png" file="KAGC_-
_Allegheny_County_Airport,_United_States_Ceiling_Height.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KAGC - Allegheny County Airport, United States Visibility" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-6e8ea7f3-11a1-4a14-989c-fe67aac7ed8f" type="image/png" file="KAGC_-
_Allegheny_County_Airport,_United_States_Visibility.png">
```

Flight Route Alerting and Briefing – API

```
<Topic name="Weather/Airport"/>
</Document>
<Document title="KPIT - Pittsburgh Intl, United States Ceiling Height" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-68c1d94f-2282-4e9c-998d-c29957fdea83" type="image/png" file="KPIT_-_Pittsburgh_Intl,_United_States_Ceiling_Height.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KPIT - Pittsburgh Intl, United States Visibility" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-533ff13f-3890-4a00-819d-3f9488bf4e42" type="image/png" file="KPIT_-_Pittsburgh_Intl,_United_States_Visibility.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KSGR - Sugar Land Regional Airport, United States Air and Dew Point Temperature" application="DTN" updateDateTime="2017-01-
31T16:01:56.744Z" authorProfile="dispatcher" id="t-e0f7d82f-0cec-4a01-96d4-2fb35c967368" type="image/png" file="KSGR_-_
_Sugar_Land_Regional_Airport,_United_States_Air_and_Dew_Point_Temperature.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KHOU - William P Hobby, United States Air and Dew Point Temperature" application="DTN" updateDateTime="2017-01-
31T16:01:56.744Z" authorProfile="dispatcher" id="t-7c4d6924-a1f9-4a95-8ec9-228c84ae08bc" type="image/png" file="KHOU_-_
_William_P_Hobby,_United_States_Air_and_Dew_Point_Temperature.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KAGC - Allegheny County Airport, United States Air and Dew Point Temperature" application="DTN" updateDateTime="2017-01-
31T16:01:56.744Z" authorProfile="dispatcher" id="t-2815bac1-1b01-40d3-8f56-d995b26f61f5" type="image/png" file="KAGC_-_
_Allegheny_County_Airport,_United_States_Air_and_Dew_Point_Temperature.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KPIT - Pittsburgh Intl, United States Air and Dew Point Temperature" application="DTN" updateDateTime="2017-01-
31T16:01:56.744Z" authorProfile="dispatcher" id="t-423ecb40-09dc-441e-bad5-a0f5be2b203c" type="image/png" file="KPIT_-_
_Pittsburgh_Intl,_United_States_Air_and_Dew_Point_Temperature.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KSGR - Sugar Land Regional Airport, United States Hourly Forecast" application="DTN" updateDateTime="2017-01-
31T16:01:56.744Z" authorProfile="dispatcher" id="t-686e59c7-9e96-4f11-b1c8-80b3022c21dd" type="image/png" file="KSGR_-_
_Sugar_Land_Regional_Airport,_United_States_Hourly_Forecast.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KSGR - Sugar Land Regional Airport, United States Hourly Forecast" application="DTN" updateDateTime="2017-01-
31T16:01:56.744Z" authorProfile="dispatcher" id="t-c30561f5-d61b-456f-b54f-cb905b462612" type="application/pdf" file="KSGR_-_
_Sugar_Land_Regional_Airport,_United_States_Hourly_Forecast.pdf">
  <Topic name="Weather/Airport"/>
</Document>
```

Flight Route Alerting and Briefing – API

```
<Document title="KHOU - William P Hobby, United States Hourly Forecast" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-b3791de1-14e7-4b99-9c02-3786a598313d" type="image/png" file="KHOU_-
_William_P_Hobby,_United_States_Hourly_Forecast.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KHOU - William P Hobby, United States Hourly Forecast" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-2771d68f-b657-4aba-95f6-0c2e82da3c40" type="application/pdf" file="KHOU_-
_William_P_Hobby,_United_States_Hourly_Forecast.pdf">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KAGC - Allegheny County Airport, United States Hourly Forecast" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-0d347016-9cbe-4dbb-8ced-788042b383df" type="image/png" file="KAGC_-
_Allegheny_County_Airport,_United_States_Hourly_Forecast.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KAGC - Allegheny County Airport, United States Hourly Forecast" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-c6bf6977-8681-4810-8201-3984b5a833a1" type="application/pdf" file="KAGC_-
_Allegheny_County_Airport,_United_States_Hourly_Forecast.pdf">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KPIT - Pittsburgh Intl, United States Hourly Forecast" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-463dce3c-af69-4b42-a610-38be1e3f464b" type="image/png" file="KPIT_-
_Pittsburgh_Intl,_United_States_Hourly_Forecast.png">
  <Topic name="Weather/Airport"/>
</Document>
<Document title="KPIT - Pittsburgh Intl, United States Hourly Forecast" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z"
authorProfile="dispatcher" id="t-65dbcea6-6331-4ca6-a740-cf7877291222" type="application/pdf" file="KPIT_-
_Pittsburgh_Intl,_United_States_Hourly_Forecast.pdf">
  <Topic name="Weather/Airport"/>
</Document>
</SubFolder>
<SubFolder title="Flight Weather">
  <Document title="Leg from 0 to 70" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z" authorProfile="dispatcher" id="t-a5a965f2-
787f-40ef-be67-75ce62ebe891" type="image/png" file="Leg_from_0_to_70.png">
    <Topic name="Weather/Chart"/>
  </Document>
  <Document title="Leg from 60 to 105" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z" authorProfile="dispatcher" id="t-51615d4a-
ef84-4b85-8e2e-160b1152dea1" type="image/png" file="Leg_from_60_to_105.png">
    <Topic name="Weather/Chart"/>
  </Document>
</SubFolder>
```


Flight Route Alerting and Briefing – API

```
<Document title="Full flight (border=1.17)" application="DTN" updateDateTime="2017-01-31T16:01:56.744Z" authorProfile="dispatcher" id="t-3848a565-627d-4961-ab2a-27c6a795c85b" type="image/png" file="Full_flight_(border=1.17).png">
  <Topic name="Weather/Chart"/>
</Document>
</SubFolder>
</SubFolder>
</EFUSUB>
```

Flight Route Alerting and Briefing – API

16 Document Revisions

Change Description	Author	Date
Updated for Other airports and add RetrieveAlerts API request.	BK	05/28/2015
Updates. Included information for demonstration application.	BK	07/07/2015
Updates. Included a section on implementation notes.	BK	07/13/2015
Updates. Included StartValidTime and EndValidTime for Other Airport.	BK	07/30/2015
Updates. Included additional web socket messages – Created, Tracked, and Untracked.	BK	08/05/2015
Updates. Included optional Climb and Descent WayPoints.	BK	09/28/2015
Updates. Fixed typo referencing number of airports.	BK	12/08/2015
Updates. Notes that flight plans are untracked at ETA. Added appendix for Alert responses.	BK	02/11/2016
Updates. Added Radar and ObservedPrecipitation Alarm type.	BK	04/11/2016
Updates. Added new request type – UpdateFlightPlanTimes.	BK	06/2/2016
Updates. Added information for Update request.	BK	07/11/2016
Fixed alarm type ‘Severe Hail’ and ‘Any Hail’ to reflect CONUS coverage rather than global	WH	08/01/2016
Added Briefing documentation for Add, Update and Retrieve Brief	GS	04/19/2017
Changed RetrieveFlightPlanSummary to RetrieveFlightPlansSummary on page 8.	MS	06/26/2017
Changed branding to DTN	WH	06/30/2017

Flight Route Alerting and Briefing – API

Defined default type values for ObservedPrecipitation when type element is not present.	BW	08/17/2017
Added StopTrackingLag element to XML examples and refresh URL parameter for refreshing state message over web sockets.	BW	08/21/2017
Modify airport and route alarm table to reflect FRA behavior.	MS	08/23/2017
Changed all mentions of Schneider Electric to DTN	WH	11/27/2017
Added alarm type for real-time Convective Turbulence, 'EDRNOWCast'	WH	12/13/2018
Inclusion of Time Buffer example	WH	02/07/2019
Changed DTN logo to new brand	WH	02/27/2019
Updated flight plan considerations	FC	03/27/2019
New time offset formats	FC	04/11/2019
Updated latitude coverage of alarms	WH	02/11/2020
Inclusion of vertical and horizontal tolerance limits	WH	03/12/2020
Added alarm type for High Ice Water Content, 'HighIceWaterContent'	WH	04/02/20
XML schema change to make 'EstimatedArrivalTime' optional	WH	06/23/20