U.S. Department of Transportation
Federal Aviation Administration

Aviation Environmental Design Tool (AEDT)

AEDT Standard Input File (ASIF) Reference Guide

Version 2a

January 2014

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
1-January-2014

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
07/13 – 1/14

4. TITLE AND SUBTITLE

Aviation Environmental Design Tool (AEDT)
AEDT Standard Input File (ASIF)
Reference Guide
Version 2a

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER
FA4TA3

5e. TASK NUMBER
KLB97

5f. WORK UNIT NUMBER

6. AUTHOR(S)

Metron Aviation
Moore, Jim
Lautman, Mark
Pepper, Jeremiah
DiFelici, John
Augustine, Stephen

U.S. DOT Volpe Center:
Koopmann, Jonathan
Solman, Gina Barberio
Ahearn, Meghan
Hwang, Sunje

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Department of Transportation
Volpe National Transportation Systems Center
55 Broadway
Cambridge, MA 02142

8. PERFORMING ORGANIZATION REPORT NUMBER

DOT-VNTSC-FAA-12-15

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Department of Transportation
Federal Aviation Administration
Office of Environment and Energy (AEE-100)
800 Independence Ave. SW
Washington, DC 20591

10. SPONSOR/MONITOR'S ACRONYM(S)
FAA

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT

This report is part of the AEDT 2a software release, and it is publicly available.

13. SUPPLEMENTARY NOTES
FAA Program Managers: Ralph Iovinelli, Rebecca Cointin, Fabio Grandi, and Christopher Sequeira

14. ABSTRACT
The Federal Aviation Administration, Office of Environment and Energy (FAA-AEE) has developed the Aviation Environmental Design Tool (AEDT) version 2a software system with the support of the following development team: FAA, National Aeronautics and Space Administration (NASA), U.S. DOT Volpe National Transportation Systems Center (Volpe Center), ATAC Corporation, Metron Aviation, Wyle Laboratories, CSSI, Inc., Foliage, MIT, and Georgia Tech. AEDT 2a is designed to dynamically model aircraft performance in space and time to compute aircraft noise, emissions, and fuel burn. In the U.S., the model is used to evaluate aircraft noise, emissions, and fuel burn of proposed air traffic airspace actions under the current version of FAA Order 1050.1E. This document provides detailed information on the AEDT Standard Input File format and requirements.

15. SUBJECT TERMS
aircraft acoustics, aircraft emissions, aircraft performance, FAA, AEDT, ASIF

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	146	Jonathan Koopmann
Unclassified	Unclassified	Unclassified			**19b. TELEPONE NUMBER (*Include area code*)** (617) 494-2885

Table of Contents

1 Introduction

The AEDT Standard Input File (ASIF) provides a standard file format to allow import of data into an AEDT study. An ASIF can be used to create new AEDT studies and to update existing AEDT studies.

This guide provides a description of the ASIF format. It also provides an overview of ASIF usage and an annotated sample study. The guide is aimed at analysts and programmers who wish to create ASIFs. It is organized into following main sections:

- Section 2 – Overview of the ASIF Format and Annotated Sample ASIF
- Section 3 – ASIF Design Considerations
- Section 4 – ASIF Requirements for Study Creation and Update
- Section 5 – ASIF XML Elements Reference Guide
- Section 6 – Procedural Profiles
- Section 7 – XML Schema – ASIF.xsd
- Section 8 – Splitting ASIF Files

2 Overview of the ASIF Format and Annotated Sample ASIF

The ASIF format allows users to import a complete AEDT study, including airports, scenarios, cases, flights, tracks, and operations. Users can also use ASIF to make partial updates to existing AEDT studies.

ASIF is based on the XML file format. XML is a text-based file format that is readable by both humans and computers. Data values are tagged with elements and organized in a hierarchical manner such that the elements can contain other elements or data. XML elements can also have attributes which provide metadata that affect how the ASIF importer processes the data in the XML file. This appendix assumes users have basic familiarity with the XML file format. For a refresher or additional information about XML, see http://xmlfiles.com/xml/.

An ASIF can be created and edited in a standard xml editor. Notepad++ is free software available for download online that can be used to create and edit XML files. A free download of Notepad++ is available at http://notepad-plus-plus.org/. The AEDT install package includes a sample ASIF you can use as a starting point to create your own ASIFs. You can find the sample ASIF at C:\AEDT\AEDT_Workspaces\ASIF_Import\Import_Files\asif_small.xml.

The FAA also supports INM and NIRS study importers to convert studies from these legacy tools directly into an ASIF.

2.1 Create New Study via ASIF

The steps to create an ASIF file in order to create a new AEDT study are as follows:

1. Create empty study file
2. Populate airport layout section
3. Create receptor set (required for studies with noise analysis)
4. Create study scenario and case hierarchy
5. Populate scenario's cases with tracks and air operations
6. Create scenario's annualization scheme

The following sections provide examples of each the above steps. For a complete explanation of the ASIF structure, see sections 4 and 5.

Please note that this is one example of an ASIF file. The naming convention used for identifiers comes from the Chicago legacy NIRS study. Each study will have its own naming convention that may differ from names used in the sample study. The ASIF sample below is provided to give context to the individual ASIF elements related to each other in an actual input file.

Step 1: Create empty study file

At a minimum, an ASIF consists of the standard XML declaration, a study section, and study metadata. In the example below, the brown text represent XML tags, and the black text represents the sample study information (which can be set by the user). Comments appear in green. For a more complete description of the ASIF elements, see section 5.

```
<?xml version="1.0" encoding="utf-8"?>
<AsifXml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.1.10"
content="study">
  <study>
    <!-- User defined study name -->
    <name>SingleFlightDeparture</name>

    <!-- Noise and Emissions is the only supported study type in AEDT 2a. -->
    <type>Noise and Emissions</type>

    <!-- Indicate the usage of metric or imperial units within the study -->
    <metricUnits>false</metricUnits>

    <!-- Emissions units to be used for the study: MetricTonnes, -->
    <!-- Kilograms, Grams, ImperialTons, or Pounds -->
    <emissionsUnits>Kilograms</emissionsUnits>

    <!-- User defined study description -->
    <description>A sample study with a single departure.</description>

    <!-- Add airport layouts here -->

    <!-- Add receptors here -->

    <!-- Add scenarios here -->

  </study>
</AsifXml>
```

Step 2: Populate airport layouts section

AEDT requires all airports in the study area to be declared. If the airport and runway are not provided in the ASIF, then AEDT will import airport and runway definitions from the AEDT system data (Airport database). In the example below, system data will be used for KORD while the defined airport elevation, location, and runways will be used for KMDW.

User-defined runways can only be created via ASIF. N-1

```
<airportLayouts>
  <layout>
    <!-- Airport with no runway tags will import runways from -->
    <!-- the AEDT system data. -->
    <airportCode type="ICAO">KORD</airportCode>
  </layout>
  <layout>
    <!-- User can specify an airport with user defined runway -->
    <airportCode type="ICAO">KMDW</airportCode>
```

```xml
    <elevation>620</elevation>
    <latitude>41.786111</latitude>
    <longitude>-87.7525</longitude>
    <!-- Airports can have one or more runways defined -->
    <runway>
      <!-- Runway length (in feet) -->
      <length>5932</length>

      <!-- Runway width (in feet) -->
      <width>150</width>

      <!-- One or more runway ends -->
      <runwayEnd>
        <!-- user defined name for runway end -->
        <name>04R</name>

        <!-- latitude and longitude of runway end -->
        <latitude>41.779496</latitude>
        <longitude>-87.75876</longitude>

        <!-- elevation in feet -->
        <elevation>0.0</elevation>

        <!-- threshold crossing height -->
        <threshCrossHeight>50.0</threshCrossHeight>

        <!-- glide slope for an approach to this runway end -->
        <glideSlope>3.0</glideSlope>

        <!-- displaced threshold for departure-->
        <depDispThresh>0.0</depDispThresh>

        <!-- displaced threshold for approach -->
        <appDispThresh>0.0</appDispThresh>

        <!-- Percent change in airport average headwind -->
        <percentWind>0.0</percentWind>
      </runwayEnd>
      <runwayEnd>
        <name>22L</name>
        <latitude>41.791167</latitude>
        <longitude>-87.743554</longitude>
        <elevation>0.0</elevation>
        <threshCrossHeight>50.0</threshCrossHeight>
        <glideSlope>3.0</glideSlope>
        <depDispThresh>0.0</depDispThresh>
        <appDispThresh>0.0</appDispThresh>
        <percentWind>0.0</percentWind>
      </runwayEnd>
    </runway>
  </layout>
</airportLayouts>
```

Step 3: Create receptor set

If the study includes noise analysis, then one or more <receptorSet> elements must be created. AEDT 2a provides two types of receptors: grid and population.

Grid receptors are used to capture environmental impacts over a large rectangular area. Receptor points are distributed evently across the grid as defined by the user.

```
<receptorSet>
  <!-- User defined name -->
  <name>gridfile_100x100</name>

  <!-- Receptor definition – either grid of centroid -->
  <grid>
    <!-- Latitude and longitude of southwest corner of grid -->
    <latitude>41.97872</latitude>
    <longitude>-87.90439</longitude>

    <!-- Width and height of grid (in nautical miles) -->
    <width>100.0</width>
    <height>100.0</height>

    <!-- Number of points across height and width of grid -->
    <numWidth>100</numWidth>
    <numHeight>100</numHeight>
  </grid>
</receptorSet>
```

Population receptors are used to capture environmental impacts at one or discrete points associated with a population count, such as US Census data.

```
<receptorSet>
  <!-- User defined name -->
  <name>pop120x160.txt</name>

  <!-- User can provide one or more population locations -->

  <centroid>
    <!-- centroid meta data, sample from US census data -->
    <stateFips>1</stateFips>
    <countyFips>1</countyFips>
    <blockId>0</blockId>
    <bnaId>0</bnaId>

    <!-- centroid location -->
    <latitude>40.642384</latitude>
    <longitude>-87.29556</longitude>

    <!-- population count at location -->
    <count>3</count>
  </centroid>

  <!-- User can add additional centroid elements here. -->

</receptorSet>
```

Step 4: Create study scenario and case hierarchy

An AEDT study is organized into scenarios. Scenarios contain a set of cases that are used to perform baseline or alternative analyses. Cases are used to group aircraft tracks and operations. Cases are used in annualization of results and during Change Analysis and Impact Evaluation.

This sample demonstrates a simple case structure similar to legacy NIRS studies. In more sophisticated studies, a case can also contain one or more cases to provide a more robust annualization tree or set of AEDT jobs.

⚠ The start time and duration of a scenario must conform to case start time and air operation time; otherwise simulation results will be invalid. W-1

```
<scenario>
  <!-- User defined scenario name and description -->
  <name>Baseline_1990</name>

  <!-- User defined start time for scenario -->
  <startTime>2009-11-10T15:02:00</startTime>

  <!-- Duration of scenario (in hours) -->
  <duration>24</duration>

  <!-- Taxi model for scenario -->
  <taxiModel>UserSpecified</taxiModel>

  <!-- Aircraft performance model -->
  <acftPerfModel>SAE1845</acftPerfModel>

  <!-- Enable/disable bank angle calculations for -->
  <!-- aircraft performance modeling -->
  <bankAngle>true</bankAngle>

  <!-- Sulfur related settings -->
  <sulfurConversionRate>0.05</sulfurConversionRate>
  <fuelSulfurContent>6.8E-4</fuelSulfurContent>

  <description>A NIRS scenario</description>

  <airportLayouts>
    <layout>KMDW</layout>
    <layout>KORD</layout>
  </airportLayouts>

  <cases>
    <!-- One or more case elements -->
    <case>
      <!-- sequential case number unique in this scenario -->
      <caseId>0</caseId>

      <!-- User defined case name -->
      <name>PlanB</name>
```

```
      <!-- Noise emissions source -->
      <source>Aircraft</source>

      <!-- Case start time and duration -->
      <startTime>2009-11-10T15:02:00</startTime>
      <duration>24</duration>

      <!-- Add trackOpSet elements here -->

    </case>
  </cases>
</scenario>
```

Step 5: Populate cases with tracks and air operations

The <trackOpSet> element defines a single track and any number of air operations to be flown on that track.

```
<!-- Add trackOpSet elements here -->
<trackOpSet>
  <!-- Single track element -->
  <track>
    <!-- User defined track id -->
    <id>T9</id>
    <!-- User defined track name -->
    <name>DJM04R_EON.10803</name>
    <!-- Track operation type: A=Arrival, D=Departure, V=Overflight -->
    <optype>D</optype>

    <!-- Airport and runway for this track -->
    <airport type="ICAO">KMDW</airport>
    <runway>04R</runway>
    <vectorCourseHelipad>0</vectorCourseHelipad>

    <!-- AEDT 2a only processes the first subtrack associated with a track -->
    <!-- dispersionWeight is not used in AEDT 2a -->
    <subtrack>
      <id>0</id>
      <dispersionWeight>1</dispersionWeight>
      <!-- Set of trackNode or trackVector elements, -->
      <!-- all must be the same for each subtrack -->

      <trackNodes>
        <trackNode>
          <latitude>40.65640</latitude>
          <longitude>-73.71322</longitude>
        </trackNode>
        <trackNode>
          <latitude>40.65640</latitude>
          <longitude>-53.71322</longitude>
        </trackNode>
      </trackNodes>
    </subtrack>
  </track>
```

```
  <!--operation element represents one or more flights on a track-->
  <operations>
    <operation>
      <!-- User defined operation id -->
      <id>T9.1</id>

      <!-- AEDT aircraftType for this operation -->
      <aircraftType>
        <airframeModel>Raytheon Beech 1900-C</airframeModel>
        <engineCode>PT67B</engineCode>
        <engineModCode>NONE</engineModCode>
      </aircraftType>

      <!-- number of times to fly this operation -->
      <numOperations>1.0</numOperations>

      <!-- User defined flight number -->
      <flightNumber>CKE545</flightNumber>

      <!-- user defined operation type -->
      <userType>MU3001</userType>
      <!-- user defined parameter data -->
      <userParam>J</userParam>

      <!-- Arrival or departure airport and runway -->
      <departureAirport type="ICAO">KMDW</departureAirport>
      <departureRunway>04R</departureRunway>
      <arrivalAirport type="FAA">LIT</arrivalAirport>

      <!-- offTime for departures or onTime for arrivals -->
      <offTime>2009-11-10T15:02:00</offTime>

      <!-- aircraft profile for this operation -->
      <saeProfile>STANDARD</saeProfile>
    </operation>
  </operations>
</trackOpSet>
```

Step 6: Create scenario's annualization tree

Annualization is the process of performing a weighted summation[1] over the noise and emission results from some or all of the cases within a scenario in order to create results that represent noise and emissions exposures over a time period of interest. Each scenario element may contain an annualization element describing the weighted annualization scheme.

```
<annualization>
  <!-- User defined scenario annualization name -->
  <name>Baseline.config</name>
  <!-- Define one or more groups of cases and groups -->
```

[1] The word 'summation' is used figuratively and the actual process of correctly summing or adding together noise or emissions results depends upon the metric being used. For example: energy metric results would not be directly added together for a result since they are logarithmic values, but would rather be log-added.

```
  <annualizationGroup>
    <weight>0.7</weight>
    <!-- Associate scenario case with this annualization group -->
    <annualizationCase>
      <!-- Specify case name to include -->
      <name>PlanB</name>
      <!-- Define rollup weight for this case -->
      <weight>1.0</weight>
    </annualizationCase>
  </annualizationGroup>
</annualization>
```

Final sample ASIF

When the previously described ASIF samples are assembled, the final version will be as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<AsifXml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.1.10"
content="study">
  <study>
    <!-- User defined study name -->
    <name>SingleFlightDeparture</name>
    <!-- Noise and Emissions is the only supported study type in AEDT 2a. -->
    <type>Noise and Emissions</type>
    <!-- Indicate the usage of metric or imperial units within the study  -->
    <metricUnits>false</metricUnits>
    <!-- Emissions units to be used for the study: MetricTonnes, -->
    <!-- Kilograms, Grams, ImperialTons, or Pounds -->
    <emissionsUnits>Kilograms</emissionsUnits>
    <!-- User defined study description -->
    <description>A sample study with a single departure.</description>
    <airportLayouts>
      <layout>
        <!-- Airport with no runway tags will import runways from -->
        <!-- the AEDT system data. -->
        <airportCode type="ICAO">KORD</airportCode>
      </layout>
      <layout>
        <!-- User can specify an airport with user defined runway -->
        <airportCode type="ICAO">KMDW</airportCode>
        <elevation>620</elevation>
        <latitude>41.786111</latitude>
        <longitude>-87.7525</longitude>

        <!-- Airports can have one or more runways defined -->
        <runway>
          <!-- Runway length (in feet) -->
          <length>5932</length>
          <!-- Runway width (in feet) -->
          <width>150</width>
          <!-- One or more runway ends -->
          <runwayEnd>
            <!-- user defined name for runway end -->
            <name>04R</name>
            <!-- latitude and longitude of runway end -->
            <latitude>41.779496</latitude>
```

```
          <longitude>-87.75876</longitude>
          <!-- elevation in feet -->
          <elevation>0.0</elevation>
          <!-- threshold crossing height -->
          <threshCrossHeight>50.0</threshCrossHeight>
          <!-- glide slope for an approach to this runway end -->
          <glideSlope>3.0</glideSlope>
          <!-- displaced threshold for departure-->
          <depDispThresh>0.0</depDispThresh>
          <!-- displaced threshold for approach -->
          <appDispThresh>0.0</appDispThresh>
          <!-- Percent change in airport average headwind -->
          <percentWind>0.0</percentWind>
        </runwayEnd>
        <runwayEnd>
          <name>22L</name>
          <latitude>41.791167</latitude>
          <longitude>-87.743554</longitude>
          <elevation>0.0</elevation>
          <threshCrossHeight>50.0</threshCrossHeight>
          <glideSlope>3.0</glideSlope>
          <depDispThresh>0.0</depDispThresh>
          <appDispThresh>0.0</appDispThresh>
          <percentWind>0.0</percentWind>
        </runwayEnd>
      </runway>
    </layout>
  </airportLayouts>
  <receptorSet>
    <!-- User defined name -->
    <name>gridfile_100x100</name>
    <!-- Receptor definition - either grid of centroid -->
    <grid>
      <!-- Latitude and longitude of southwest corner of grid -->
      <latitude>41.97872</latitude>
      <longitude>-87.90439</longitude>
      <!-- Width and height of grid (in nautical miles) -->
      <width>100.0</width>
      <height>100.0</height>
      <!-- Number of points across height and width of grid -->
      <numWidth>100</numWidth>
      <numHeight>100</numHeight>
    </grid>
  </receptorSet>
  <receptorSet>
    <!-- User defined name -->
    <name>pop120x160.txt</name>

    <!-- User can provide one or more population locations -->

    <centroid>
      <!-- centroid meta data, sample from US census data -->
      <stateFips>1</stateFips>
      <countyFips>1</countyFips>
      <blockId>0</blockId>
      <bnaId>0</bnaId>
```

```
        <!-- centroid location -->
        <latitude>40.642384</latitude>
        <longitude>-87.29556</longitude>

        <!-- population count at location -->
        <count>3</count>
      </centroid>

      <!-- User can add additional centroid elements here. -->

    </receptorSet>

    <scenario>
      <!-- User defined scenario name and description -->
      <name>Baseline_1990</name>
      <!-- User defined start time for scenario -->
      <startTime>2009-11-10T15:02:00</startTime>
      <!-- Duration of scenario (in hours) -->
      <duration>24</duration>
      <!-- Taxi model for scenario -->
      <taxiModel>UserSpecified</taxiModel>
      <!-- Aircraft performance model -->
      <acftPerfModel>SAE1845</acftPerfModel>
      <!-- Enable/disable bank angle calculations for aircraft -->
      <!-- performance modeling -->
      <bankAngle>true</bankAngle>
      <!-- Sulfur related settings -->
      <sulfurConversionRate>0.05</sulfurConversionRate>
      <fuelSulfurContent>6.8E-4</fuelSulfurContent>
      <description>A NIRS scenario</description>
      <airportLayouts>
        <layout>KMDW</layout>
        <layout>KORD</layout>
      </airportLayouts>
      <cases>
        <!-- One or more case elements -->
        <case>
          <!-- -->
          <caseId>0</caseId>
          <!-- User defined case name -->
          <name>PlanB</name>
          <!-- Noise emissions source -->
          <source>Aircraft</source>
          <!-- Case start time and duration -->
          <startTime>2009-11-10T15:02:00</startTime>
          <duration>24</duration>
          <!-- Add trackOpSet elements here -->
          <trackOpSet>
            <!-- Single track element -->
            <track>
              <!-- User defined track id -->
              <id>T9</id>
              <!-- User defined track name -->
              <name>DJM04R_EON.10803</name>
              <!-- Track operation type: A = Arrival, D = Departure, -->
              <!-- V = Overflight -->
```

```
            <optype>D</optype>
            <!-- Airport and runway for this track -->
            <airport type="ICAO">KMDW</airport>
            <runway>04R</runway>
            <vectorCourseHelipad>0</vectorCourseHelipad>
            <!-- AEDT 2a only processes the first subtrack associated with a track -->
            <!-- dispersionWeight is not used in AEDT 2a -->
            <subtrack>
              <id>0</id>
              <dispersionWeight>1</dispersionWeight>
              <!-- Set of trackNode or trackVector elements, -->
              <!-- all must be the same for each subtrack-->
              <trackNodes>
                <trackNode>
                  <latitude>40.65640</latitude>
                  <longitude>-73.71322</longitude>
                </trackNode>
                <trackNode>
                  <latitude>40.65640</latitude>
                  <longitude>-53.71322</longitude>
                </trackNode>
              </trackNodes>
            </subtrack>
          </track>
          <!-- operations represents one or more flights on a track -->
          <operations>
            <operation>
              <!-- User defined operation id -->
              <id>T9.1</id>
              <!-- aircraftType for this operation -->
              <aircraftType>
                <airframeModel>Raytheon Beech 1900-C</airframeModel>
                <engineCode>PT67B</engineCode>
                <engineModCode>NONE</engineModCode>
              </aircraftType>
              <!-- number of times to fly this operation -->
              <numOperations>1.0</numOperations>
              <!-- User defined flight number -->
              <flightNumber>CKE545</flightNumber>
              <!-- user defined operation type -->
              <userType>MU3001</userType>
              <!-- user defined parameter data -->
              <userParam>J</userParam>
              <!-- Arrival or departure airport and runway -->
              <departureAirport type="ICAO">KMDW</departureAirport>
              <departureRunway>04R</departureRunway>
              <arrivalAirport type="FAA">LIT</arrivalAirport>
              <!-- offTime for departures or onTime for arrivals -->
              <offTime>2009-11-10T15:02:00</offTime>
              <!-- aircraft profile for this operation -->
              <saeProfile>STANDARD</saeProfile>
            </operation>
          </operations>
        </trackOpSet>
      </case>
    </cases>
```

```xml
      <annualization>
        <!-- User defined scenario annualization name -->
        <name>Baseline.config</name>
        <!-- Define one or more groups of cases and groups -->
        <annualizationGroup>
          <weight>0.7</weight>
          <!-- Associate scenario case with this annualization group -->
          <annualizationCase>
            <!-- Specify case name to include -->
            <name>PlanB</name>
            <!-- Define rollup weight for this case -->
            <weight>1.0</weight>
          </annualizationCase>
        </annualizationGroup>
      </annualization>
    </scenario>
  </study>
</AsifXml>
```

3 ASIF Design Considerations

3.1 Event Consolidation

AEDT calculates noise for all air operations (e.g. all instances of an aircraft and track) in a given case, which differs from the legacy tool, NIRS. In order to optimize noise modeling performance in AEDT it is suggested to combine like operations in a case into a representative single air operation for entry into the ASIF. See the AEDT User Guide for instructions on how to combine like entries from an annual schedule.

Sample ASIF using discrete air operations:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<AsifXml xmlns:AsifXml="http://www.faa.gov/ASIF"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.1.10" content="study">
  <study xmlns:asif="http://www.faa.gov/ASIF">
    <name>Chicago70</name>
    <type>Noise and Emissions</type>
    <metricUnits>false</metricUnits>
    <emissionsUnits>Kilograms</emissionsUnits>
    <description>A sample NIRS study</description>
    <boundary>
      <polygon>
        <vertex>
          <latitude>40.636993970695244</latitude>
          <longitude>-89.21758333055047</longitude>
        </vertex>
        <vertex>
          <latitude>40.636993970695244</latitude>
          <longitude>-86.59119444944956</longitude>
        </vertex>
        <vertex>
          <latitude>43.3047921364604</latitude>
          <longitude>-86.53522348936178</longitude>
        </vertex>
        <vertex>
          <latitude>43.3047921364604</latitude>
          <longitude>-89.27355429063823</longitude>
        </vertex>
      </polygon>
    </boundary>
    <airportLayouts>
      <layout>
        <airportCode>ORD</airportCode>
        <runway>
          <length>7878</length>
          <width>150</width>
          <runwayEnd>
            <name>09L</name>
            <latitude>41.984329</latitude>
            <longitude>-87.918854</longitude>
            <elevation>659.7</elevation>
            <threshCrossHeight>50.0</threshCrossHeight>
```

```
          <glideSlope>3.0</glideSlope>
          <depDispThresh>0.0</depDispThresh>
          <appDispThresh>0.0</appDispThresh>
          <percentWind>0.0</percentWind>
        </runwayEnd>
        <runwayEnd>
          <name>27R</name>
          <latitude>41.984406</latitude>
          <longitude>-87.889786</longitude>
          <elevation>650.0</elevation>
          <threshCrossHeight>50.0</threshCrossHeight>
          <glideSlope>3.0</glideSlope>
          <depDispThresh>0.0</depDispThresh>
          <appDispThresh>0.0</appDispThresh>
          <percentWind>0.0</percentWind>
        </runwayEnd>
      </runway>
    </layout>
  </airportLayouts>
  <scenario>
    <name>Baseline_1990</name>
    <startTime>2011-10-25T00:00:00</startTime>
    <duration>24</duration>
    <taxiModel>UserSpecified</taxiModel>
    <acftPerfModel>SAE1845</acftPerfModel>
    <bankAngle>true</bankAngle>
    <altitudeCutoff>18723.1</altitudeCutoff>
    <sulfurConversionRate>0.05</sulfurConversionRate>
    <fuelSulfurContent>6.8E-4</fuelSulfurContent>
    <description>Sample Scenario – discrete air ops </description>
    <airportLayouts>
      <layout>ORD</layout>
    </airportLayouts>
    <cases>
      <case>
        <caseId>0</caseId>
        <name>PlanB</name>
        <source>Aircraft</source>
        <startTime>2011-10-25T00:00:00</startTime>
        <duration>24</duration>
        <case>
          <caseId>6</caseId>
          <name>exexbf2.ord.day.dep_STD</name>
          <source>Aircraft</source>
          <startTime>2011-10-25T00:00:00</startTime>
          <duration>24</duration>
          <description>Sample Case – discrete air operations</description>
          <trackOpSet>
            <track>
              <id>T0.N</id>
              <name>D9LJ_GIJ_and_No_route_name</name>
              <optype>D</optype>
              <airport>ORD</airport>
              <runway>09L</runway>
              <subtrack>
                <id>0</id>
```

```
<dispersionWeight>1.0</dispersionWeight>
<trackNodes>
  <trackNode>
    <id>RWY_09L</id>
    <latitude>41.984329</latitude>
    <longitude>-87.918854</longitude>
  </trackNode>
  <trackNode>
    <id>DORD_09L@0</id>
    <latitude>41.984406</latitude>
    <longitude>-87.889786</longitude>
  </trackNode>
  <trackNode>
    <id>DORD9L2@20</id>
    <latitude>41.984043</latitude>
    <longitude>-87.801483</longitude>
  </trackNode>
  <trackNode>
    <id>DORD9L3@30</id>
    <latitude>41.984043</latitude>
    <longitude>-87.724838</longitude>
  </trackNode>
  <trackNode>
    <id>DORD22L9L@50</id>
    <latitude>41.982971</latitude>
    <longitude>-87.592621</longitude>
  </trackNode>
  <trackNode>
    <id>DGIJJ@80</id>
    <latitude>41.976048</latitude>
    <longitude>-87.303543</longitude>
  </trackNode>
  <trackNode>
    <id>DGIJJ2@150</id>
    <latitude>41.974567</latitude>
    <longitude>-86.741348</longitude>
  </trackNode>
  <trackNode>
    <id>GIJLB@210</id>
    <latitude>42.03212</latitude>
    <longitude>-87.152512</longitude>
  </trackNode>
  <trackNode>
    <id>GIJ@210</id>
    <latitude>41.76833</latitude>
    <longitude>-86.318329</longitude>
  </trackNode>
  <trackNode>
    <id>MODEM@270</id>
    <latitude>41.7225</latitude>
    <longitude>-84.89917</longitude>
  </trackNode>
  <trackNode>
    <id>GERBS@270</id>
    <latitude>41.7775</latitude>
    <longitude>-84.425003</longitude>
```

```
          </trackNode>
          <trackNode>
            <id>CETUS@270</id>
            <latitude>41.703609</latitude>
            <longitude>-82.81694</longitude>
          </trackNode>
        </trackNodes>
      </subtrack>
    </track>
    <operations>
      <operation>
        <id>T1.0</id>
        <aircraftType>
          <anpAircraftId>757PW</anpAircraftId>
        </aircraftType>
        <numOperations>1.0</numOperations>
        <flightNumber>UA_1231</flightNumber>
        <userType>757PW</userType>
        <userParam>J</userParam>
        <departureAirport>ORD</departureAirport>
        <departureRunway>09L</departureRunway>
        <arrivalAirport>DTW</arrivalAirport>
        <offTime>2011-10-25T11:00:00</offTime>
        <saeProfile>STANDARD</saeProfile>
      </operation>
      <operation>
        <id>T1.2</id>
        <aircraftType>
          <anpAircraftId>757PW</anpAircraftId>
        </aircraftType>
        <numOperations>1.0</numOperations>
        <flightNumber>CD_2282</flightNumber>
        <userType>757PW</userType>
        <userParam>J</userParam>
        <departureAirport>ORD</departureAirport>
        <departureRunway>09L</departureRunway>
        <arrivalAirport>DTW</arrivalAirport>
        <offTime>2011-10-25T12:00:00</offTime>
        <saeProfile>STANDARD</saeProfile>
      </operation>
      <operation>
        <id>T1.3</id>
        <aircraftType>
          <anpAircraftId>757PW</anpAircraftId>
        </aircraftType>
        <numOperations>1.0</numOperations>
        <flightNumber>CD_2283</flightNumber>
        <userType>757PW</userType>
        <userParam>J</userParam>
        <departureAirport>ORD</departureAirport>
        <departureRunway>09L</departureRunway>
        <arrivalAirport>DTW</arrivalAirport>
        <offTime>2011-10-25T13:00:00</offTime>
        <saeProfile>STANDARD</saeProfile>
      </operation>
    </operations>
```

```
            </trackOpSet>
          </case>
        </case>
      </cases>
    </scenario>
  </study>
</AsifXml>
```

Consolidated version of above ASIF:

```
<?xml version="1.0" encoding="utf-8" ?>
<AsifXml xmlns:AsifXml="http://www.faa.gov/ASIF"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.1.10" content="study">
  <study xmlns:asif="http://www.faa.gov/ASIF">
    <name>Chicago70</name>
    <type>Noise and Emissions</type>
    <metricUnits>false</metricUnits>
    <emissionsUnits>Kilograms</emissionsUnits>
    <description>Example of rolled up air operations</description>
    <boundary>
      <polygon>
        <vertex>
          <latitude>40.636993970695244</latitude>
          <longitude>-89.21758333055047</longitude>
        </vertex>
        <vertex>
          <latitude>40.636993970695244</latitude>
          <longitude>-86.59119444944956</longitude>
        </vertex>
        <vertex>
          <latitude>43.3047921364604</latitude>
          <longitude>-86.53522348936178</longitude>
        </vertex>
        <vertex>
          <latitude>43.3047921364604</latitude>
          <longitude>-89.27355429063823</longitude>
        </vertex>
      </polygon>
    </boundary>
    <airportLayouts>
      <layout>
        <airportCode>ORD</airportCode>
        <runway>
          <length>7878</length>
          <width>150</width>
          <runwayEnd>
            <name>09L</name>
            <latitude>41.984329</latitude>
            <longitude>-87.918854</longitude>
            <elevation>659.7</elevation>
            <threshCrossHeight>50.0</threshCrossHeight>
            <glideSlope>3.0</glideSlope>
            <depDispThresh>0.0</depDispThresh>
            <appDispThresh>0.0</appDispThresh>
            <percentWind>0.0</percentWind>
          </runwayEnd>
          <runwayEnd>
```

```
          <name>27R</name>
          <latitude>41.984406</latitude>
          <longitude>-87.889786</longitude>
          <elevation>650.0</elevation>
          <threshCrossHeight>50.0</threshCrossHeight>
          <glideSlope>3.0</glideSlope>
          <depDispThresh>0.0</depDispThresh>
          <appDispThresh>0.0</appDispThresh>
          <percentWind>0.0</percentWind>
        </runwayEnd>
      </runway>
    </layout>
  </airportLayouts>
  <scenario>
    <name>Baseline_1990</name>
    <startTime>2011-10-25T00:00:00</startTime>
    <duration>24</duration>
    <taxiModel>UserSpecified</taxiModel>
    <acftPerfModel>SAE1845</acftPerfModel>
    <bankAngle>true</bankAngle>
    <altitudeCutoff>18723.1</altitudeCutoff>
    <sulfurConversionRate>0.05</sulfurConversionRate>
    <fuelSulfurContent>6.8E-4</fuelSulfurContent>
    <description>Sample Scenario – rolled up air ops</description>
    <airportLayouts>
      <layout>ORD</layout>
    </airportLayouts>
    <cases>
      <case>
        <caseId>0</caseId>
        <name>PlanB</name>
        <source>Aircraft</source>
        <startTime>2011-10-25T00:00:00</startTime>
        <duration>24</duration>
        <case>
          <caseId>6</caseId>
          <name>exexbf2.ord.day.dep_STD</name>
          <source>Aircraft</source>
          <startTime>2011-10-25T00:00:00</startTime>
          <duration>24</duration>
          <description>Rolled up operations</description>
          <trackOpSet>
            <track>
              <id>T0.N</id>
              <name>D9LJ_GIJ_and_No_route_name</name>
              <optype>D</optype>
              <airport>ORD</airport>
              <runway>09L</runway>
              <subtrack>
                <id>0</id>
                <dispersionWeight>1.0</dispersionWeight>
                <trackNodes>
                  <trackNode>
                    <id>RWY_09L</id>
                    <latitude>41.984329</latitude>
                    <longitude>-87.918854</longitude>
```

```
</trackNode>
<trackNode>
  <id>DORD_09L@0</id>
  <latitude>41.984406</latitude>
  <longitude>-87.889786</longitude>
</trackNode>
<trackNode>
  <id>DORD9L2@20</id>
  <latitude>41.984043</latitude>
  <longitude>-87.801483</longitude>
</trackNode>
<trackNode>
  <id>DORD9L3@30</id>
  <latitude>41.984043</latitude>
  <longitude>-87.724838</longitude>
</trackNode>
<trackNode>
  <id>DORD22L9L@50</id>
  <latitude>41.982971</latitude>
  <longitude>-87.592621</longitude>
</trackNode>
<trackNode>
  <id>DGIJJ@80</id>
  <latitude>41.976048</latitude>
  <longitude>-87.303543</longitude>
</trackNode>
<trackNode>
  <id>DGIJJ2@150</id>
  <latitude>41.974567</latitude>
  <longitude>-86.741348</longitude>
</trackNode>
<trackNode>
  <id>GIJLB@210</id>
  <latitude>42.03212</latitude>
  <longitude>-87.152512</longitude>
</trackNode>
<trackNode>
  <id>GIJ@210</id>
  <latitude>41.76833</latitude>
  <longitude>-86.318329</longitude>
</trackNode>
<trackNode>
  <id>MODEM@270</id>
  <latitude>41.7225</latitude>
  <longitude>-84.89917</longitude>
</trackNode>
<trackNode>
  <id>GERBS@270</id>
  <latitude>41.7775</latitude>
  <longitude>-84.425003</longitude>
</trackNode>
<trackNode>
  <id>CETUS@270</id>
  <latitude>41.703609</latitude>
  <longitude>-82.81694</longitude>
</trackNode>
```

```xml
                  </trackNodes>
                </subtrack>
              </track>
              <operations>
                <operation>
                  <id>T1.0-3.0</id>
                  <aircraftType>
                    <anpAircraftId>757PW</anpAircraftId>
                  </aircraftType>
                  <numOperations>3.0</numOperations>
                  <flightNumber>Rollup1</flightNumber>
                  <userType>757PW</userType>
                  <userParam>J</userParam>
                  <departureAirport>ORD</departureAirport>
                  <departureRunway>09L</departureRunway>
                  <arrivalAirport>DTW</arrivalAirport>
                  <offTime>2011-10-25T11:00:00</offTime>
                  <saeProfile>STANDARD</saeProfile>
                </operation>
              </operations>
            </trackOpSet>
          </case>
        </case>
      </cases>
    </scenario>
  </study>
</AsifXml>
```

3.2 Event Dates for Average Annual Day Study

For a study with average annual day data events, scenarios, and cases must all be entered with the same date.

3.3 Aircraft Tagging with User Defined Aircraft

Tracking a specific aircraft type can be done by creating a duplicate of the aircraft of interest and naming it with a unique name. An example ASIF fragment that can be added to the <fleet> element of an ASIF is provided below:

```xml
    <airplane>
      <description>B737-200/JT8D-17 NORDAM B737 LGW HUSHKIT</description>
      <baseAirplane>
        <anpAirplaneId>737N17</anpAirplaneId>
        <badaAirplaneId>B732</badaAirplaneId>
        <airframeModel>Boeing 737-200 Series</airframeModel>
        <engineCode>1PW010</engineCode>
        <engineModCode>NORH</engineModCode>
      </baseAirplane>
      <anpAirplaneInfo>
        <anpAirplane>
          <!-- Create a unique anpAirplaneId to use for targeted air operations -->
          <anpAirplaneId>737N17-U</anpAirplaneId>
          <!-- Create a unique description to use for targeted air operations -->
          <description>Targeted B737-200/JT8D-17</description>
```

```
        <maxGrossWeightTakeoff>124000</maxGrossWeightTakeoff>
        <maxGrossWeightLand>107000</maxGrossWeightLand>
        <maxDsStop>4244</maxDsStop>
        <thrustStatic>16000</thrustStatic>
        <noiseId>2JT8DN</noiseId>
      </anpAirplane>
    </anpAirplaneInfo>
    <airframeInfo>
      <airframeModel>Boeing 737-200 Series</airframeModel>
    </airframeInfo>
  </airplane>
```

3.4 Number of Operations in a Case and Results Reuse

AEDT has the ability to reuse previously calculated results when running a new job. The smallest unit of results that can be reused is a set of air operations in a case. Run time can be optimized by designing the ASIF with this capability in mind.

3.5 Control Codes in AEDT

3.5.1 Altitude Controls

AEDT will fly AtOrBelow control codes as close to the specified altitude as possible, which differs from the legacy tool NIRS that accepts any altitude at or below the specified altitude. Similarly, AEDT will fly AtOrAbove control codes as close to the specified altitude as possible while NIRS accepts any altitude at or above the specified altitude.

AEDT will not use control codes below 500 ft. AFE. Since NIRS does not use control codes below 3000 ft. AFE, any NIRS control codes that are converted to ASIF that are at or below 3000 feet should be changed to the AEDT AtOrBelow control code.

When modeling runway to runway operations using sensor path data, define the flight path using the ASIF sensorPath element rather than the track element. Sensor paths provide more direct control of altitude for an aircraft trajectory.

AEDT will fly the length of ground tracks without requiring altitude control codes at the beginning and end of the tracks. See 5.2.50 on defining control codes (nodeControlType).

3.5.2 Speed Controls

Although the AEDT ASIF schema allows for the definion of speed controls, these are not utilized in AEDT 2a and will be ignored.

3.6 Modeling Intra-Study Flights (Runway to Runway)

Intra-Study flights have arrival and departure airports that are inside the study area. These flights are modeled in the legacy tool NIRS as a pair of arrival and departure operations touching in the middle (e.g. the non-runway ends meet). AEDT supports this modeling technique.

However, when the Truncate and/or Extend Flight Paths to Study Boundary checkbox in the *Run Options* dialog box is invoked, AEDT will extend the non-runway end of these tracks to intersect the study boundary. This can cause AEDT to add undesired trajectory segments for intra-study flights modeled with a pair of operations. It is recommended to use sensor path data to define a continuous flight path for intra-study (runway to runway) flights when using the Truncate and/or Extend Flight Paths to Study Boundary option in AEDT 2a to prevent the addition of undesired trajectory segments.

Results from intra-study flight (runway to runway) air operations that are based on sensor path data (rather than profiles and/or tracks) will include some segments between the terminal area portion (10,000ft or higher above field elevation) and the cruise portion (above the nominal cruise altitude of the sensor data) that are modeled as though they were part of the cruise portion. That is, the trajectory mode listed in flight performance reports for such segments will be Cruise, rather than EnrouteClimb or EnrouteDescent. The fuel burn for these segments will be based on the BADA fuel burn model for cruise, rather than the model for climb or descent, and emissions results that depend on fuel burn are affected accordingly.

4 ASIF Requirements for Study Creation and Update

There are two types of ASIF import files: a full-study import and a partial-study import. The following sections describe each type of import file.

4.1 Create New Study with ASIF

AEDT supports the creation of new studies via ASIF. For a full-study import, the *content* attribute of the <AsifXML> element must be set to "study".

The table below illustrates the hierarchical relationship of structural XML elements within the ASIF import file; some elements are optional. For detailed information about a particular element, refer to the corresponding listed element reference.

XML Hierarchy for Full Study Import

ASIF Element	Element Reference
<AsifXml>	Section 5.2.22
<options>	Section 5.2.37
<utmZoneDefault>	Section 5.2.37
<study>	Section 5.2.47
<name>	Section 5.2.47
<type>	Section 5.2.47
<metricUnits>	Section 5.2.47
<emissionsUnits>	Section 5.2.47
<description>	Section 5.2.47
<boundary>	Section 5.2.28
<polygon>	Section 5.2.38
<vertex>	Section 5.2.57
<latitude>	Section 5.3.3
<latitudeDMS>	Section 5.3.3
<longitude>	Section 5.3.3
<longitudeDMS>	Section 5.3.3
<utmN>	Section 5.3.6
<utmE>	Section 5.3.6
<utmZone>	Section 5.3.6
<climate>	Section 5.2.32

ASIF Element	Element Reference
<identifier>	Section 5.2.32
<temperature>	Section 5.2.32
<pressure>	Section 5.2.32
<humidity>	Section 5.2.32
<headWind>	Section 5.2.32
<seaLevelPressure>	Section 5.2.32
<dewPoint>	Section 5.2.32
<windDirection>	Section 5.2.32
<visibility>	Section 5.2.32
<airportLayouts>	Section 5.2.7
<layout>	Section 5.2.6
<name>	Section 5.2.6
<airportCode>	Section 5.2.5
<type>	Section 5.2.5
<country>	Section 5.2.5
<startDate>	Section 5.2.6
<runway>	Section 5.2.42
<length>	Section 5.2.42
<width>	Section 5.2.42
<runwayEnd>	Section 5.2.43
<name>	Section 5.2.43
<latitude>	Section 5.3.3
<latitudeDMS>	Section 5.3.3
<longitude>	Section 5.3.3
<longitudeDMS>	Section 5.3.3
<utmN>	Section 5.3.6
<utmE>	Section 5.3.6
<utmZone>	Section 5.3.6
<elevation>	Section 5.2.43
<threshCrossHeight>	Section 5.2.43

ASIF Element	Element Reference
<threshElevation>	Section 5.2.43
<glideSlope>	Section 5.2.43
<intAltitude>	Section 5.2.43
<depDispThresh>	Section 5.2.43
<appDispThresh>	Section 5.2.43
<percentWind>	Section 5.2.43
<isHelipad>	Section 5.2.43
<tracks>	Section 5.2.54
<terrainFiles>	Section 5.2.47
<receptorSet>	Section 5.2.40
<name>	Section 5.2.40
<latitude>	Section 5.3.3
<latitudeDMS>	Section 5.3.3
<longitude>	Section 5.3.3
<longitudeDMS>	Section 5.3.3
<centroid>	Section 5.2.31
<stateFips>	Section 5.2.31
<countyFips>	Section 5.2.31
<blockId>	Section 5.2.31
<bnaId>	Section 5.2.31
<latitude>	Section 5.3.3
<latitudeDMS>	Section 5.3.3
<longitude>	Section 5.3.3
<longitudeDMS>	Section 5.3.3
<utmN>	Section 5.3.6
<utmE>	Section 5.3.6
<utmZone>	Section 5.3.6
<elevation>	Section 5.2.31
<count>	Section 5.2.31
<grid>	Section 5.2.34

ASIF Element	Element Reference
<latitude>	Section 5.3.3
<latitudeDMS>	Section 5.3.3
<longitude>	Section 5.3.3
<longitudeDMS>	Section 5.3.3
<utmN>	Section 5.3.6
<utmE>	Section 5.3.6
<utmZone>	Section 5.3.6
<elevation>	Section 5.2.34
<width>	Section 5.2.34
<height>	Section 5.2.34
<numWidth>	Section 5.2.34
<numHeight>	Section 5.2.34
<fleet>	Section 5.2.33
<anpNoiseGroup>	Section 5.2.15
<noiseId>	Section 5.2.15
<spectralClassApproach>	Section 5.2.15
<spectralClassDeparture>	Section 5.2.15
<spectralClassAfterburner>	Section 5.2.15
<thrustSetType>	Section 5.2.15
<modelType>	Section 5.2.15
<npdCurves>	Section 5.2.15
<airplane>	Section 5.2.4
<description>	Section 5.2.4
<baseAirplane>	Section 5.2.27
<anpAirplaneInfo>	Section 5.2.12
<anpAirplaneId>	Section 5.2.12
<anpAirplane>	Section 5.2.11
<scenario>	Section 5.2.44
<name>	Section 5.2.44
<startTime>	Section 5.2.44

ASIF Element	Element Reference
<duration>	Section 5.2.44
<taxiModel>	Section 5.2.44
<acftPerfModel>	Section 5.2.44
<bankAngle>	Section 5.2.44
<altitudeCutoff>	Section 5.2.44
<sulfurConversionRate>	Section 5.2.44
<fuelSulfurContent>	Section 5.2.44
<description>	Section 5.2.44
<airportLayouts>	Section 5.2.7
<name>	Section 5.2.6
<airportCode>	Section 5.2.5
<type>	Section 5.2.5
<country>	Section 5.2.5
<startDate>	Section 5.2.6
<runway>	Section 5.2.42
<length>	Section 5.2.42
<width>	Section 5.2.42
<runwayEnd>	Section 5.2.43
<name>	Section 5.2.43
<latitude>	Section 5.3.3
<latitudeDMS>	Section 5.3.3
<longitude>	Section 5.3.3
<longitudeDMS>	Section 5.3.3
<utmN>	Section 5.3.6
<utmE>	Section 5.3.6
<utmZone>	Section 5.3.6
<elevation>	Section 5.2.43
<threshCrossHeight>	Section 5.2.43
<threshElevation>	Section 5.2.43
<glideSlope>	Section 5.2.43

ASIF Element	Element Reference
<intAltitude>	Section 5.2.43
<depDispThresh>	Section 5.2.43
<appDispThresh>	Section 5.2.43
<percentWind>	Section 5.2.43
<isHelipad>	Section 5.2.43
<cases>	Section 5.2.30
<case>	Section 5.2.29
<caseId>	Section 5.2.29
<name>	Section 5.2.29
<source>	Section 5.2.29
<startTime>	Section 5.2.29
<duration>	Section 5.2.29
<climateId>	Section 5.2.29
<hourlyWxFile>	Section 5.2.29
<hourlyWxMD5>	Section 5.2.29
<description>	Section 5.2.29
<case>	Section 5.2.29
<trackOpSet>	Section 5.2.52
<operation>	Section 5.2.35
<reference>	Section 5.2.41
<annualization>	Section 5.2.8
<name>	Section 5.2.8
<annualizationGroup>	Section 5.2.10
<weight>	Section 5.2.10
<scaleFactor>	Section 5.2.10
<annualizationGroup>	Section 5.2.10
<annualizationCase>	Section 5.2.9
<name>	Section 5.2.9
<weight>	Section 5.2.9
<scaleFactor>	Section 5.2.9

4.2 Update Existing Study via ASIF

ASIFs can be used to update existing studies. An ASIF study update file contains a portion of an AEDT study; the remaining required elements must already exist in the project to which you are importing the file.

The table below illustrates the hierarchical relationship of structural XML elements within a partial import ASIF. Partial-study import ASIFs may contain an optional `<options>` element and can only contain a single content element type from the table below. In addition, the *content* attribute of the <AsifXML> element must be set to the corresponding element type. A sample partial ASIF shown below contains only the receptor set data. Note that the *content* attribute of the <AsifXML> element is set to "receptorSets".

Each study update element has an assigned import context. AEDT will use the ASIF import context to determine where to add the new data in the existing study. An import context can be one of the following:

- Study—element is a study-level element and does not require prompting the user for an import context
- Scenario—element must be associated with an existing scenario in the study, and the user is prompted to select a scenario
- Case—element must be associated with an existing case in the study, and the user is prompted to select a scenario and related case

For detailed information about a particular element, click on the associated link.

XML Hierarchy for Partial Study Import

ASIF Element	Import Context	Element Reference
`<AsifXml>`	n/a	Section 5.2.22
`<options>`	study	Section 5.2.37
`<airportLayouts>`	study	Section 5.2.6
`<annualization>`	scenario	Section 5.2.8
`<boundary>`	study	Section 5.2.28
`<case>`	scenario	Section 5.2.29
`<fleet>`	study	Section 5.2.33
`<operation>`	case	Section 5.2.35
`<receptorSet>`	study	Section 5.2.40
`<scenario>`	study	Section 5.2.44
`<trackOpSet>`	case	Section 5.2.52

Sample Partial ASIF – Receptor Set


```xml
<?xml version="1.0" encoding="utf-8"?>
<AsifXml version="1.1.10" content="receptorSets" xmlns:AsifXml="http://www.faa.gov/ASIF"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <receptorSet>
    <name>Partial ASIF Receptor Set</name>
    <centroid>
      <stateFips>17</stateFips>
      <countyFips>031</countyFips>
      <blockId>1</blockId>
      <bnaId>0</bnaId>
      <latitude>40.642384</latitude>
      <longitude>-87.29556</longitude>
      <count>3</count>
    </centroid>
    <centroid>
      <stateFips>17</stateFips>
      <countyFips>031</countyFips>
      <blockId>2</blockId>
      <bnaId>1</bnaId>
      <latitude>41.074768</latitude>
      <longitude>-87.259346</longitude>
      <count>11</count>
    </centroid>
  </receptorSet>
</AsifXml>
```

5 ASIF XML Elements Reference Guide

5.1 Terminology Used in the Schema

This section describes notation used in the schema. The next two tables describe the notation for XML tag types and the notation for required number of elements.

Notation for ASIF XML Tag Types

Type	Description
integer, float, double, boolean	The standard numeric types
boolean	T, true, or 1 for TRUE values, and F, false, or 0 for FALSE values.
stringN	A string with up to N characters.
datetime	A date and time string of the format YYYY-MM-DD HH:mm:SS.sss **YYYY :** four digit year **MM :** two digit month (from 01-12) **DD :** two digit day of the month (from 01 to last day) **HH :** two digit hour (from 00-23) **mm :** two digit minutes (from 00-59) **SS :** (optional) seconds (from 00-59) **sss :** (optional) milliseconds (from 000-999)
enum	An enumeration. See the element's description for valid values.
G	A group type. This type indicates that the actual element tag is really a placeholder for a group of element tags. XML tags with the group type will be italicized.
S	Special. See the element description for details.
-	A complex type that contains other elements.

Notation for the Required Number of Elements

Num	Description
+	1 or more instances are required.
*	0 or more instances are required, implying the element is optional if 0 elements are desired.
?	0 or 1 instance is required, again implying an optional element.
N	N instances are required, where N is a positive integer.
N+	N or more instances are required, where N is a positive integer.

Num	Description
S	In some cases, the requirement of an element (or group of elements) may depend on special circumstances, in which case the element (or group of elements) will be marked with an S and the specific requirements will be detailed in the description section.

Some element descriptions include a Choice column. This column indicates you need to choose between one of the elements associated with the same choice letter. For example, referring to the table in section 5.3.3, choice '`a`' refers to a choice between the `latitude` and `latitudeDMS` elements, and choice '`b`' refers to the `longitude` and `longitudeDMS` elements. When creating a tag of type `latlonCoordGroup`, you can include one element from choice '`a`', and one element from choice '`b`'.

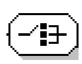
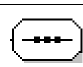
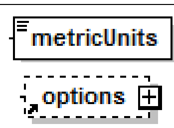
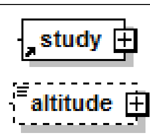
Some ASIF elements contain attributes. For example, when specifying an airport, you can include the `airportCode` element. This element has a `type` attribute which indicates the type of airport code. In the example below, the `type` attribute indicates that the type of airport code is ICAO:

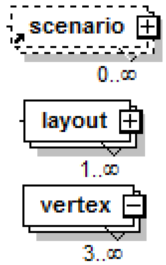
```
<airportCode type="ICAO">KMDW</airportCode>
```

Section 5.2 describes attributes when they are defined for a particular element.

Notation for Schema Diagram

The schema diagram illustrates the structure and contents of each XML element. It facilitates understanding the relationship between XML elements, and the rules and properties of each element. The following table lists the notations used in the diagrams.

Notation	Sample Icon	Description
Choice indicator		Only one of the elements contained in the selected group can be present
Sequence indicator		Child elements must appear in the specified sequence
Element		Represented by a rectangle with solid or dotted border • Solid rectangle – required element • Dotted rectangle – optional element
Element with (+) sign		Indicates that the element has child element(s) and/or attribute(s)

Notation	Sample Icon	Description
Element with min and max bound		Specifies the min/max number of times an element can occur in the parent element

XML elements in ASIF must be in the order as specified in the ASIF schema. N-2

5.2 Element Descriptions

5.2.1 aircraftType

Characterizes an aircraft.

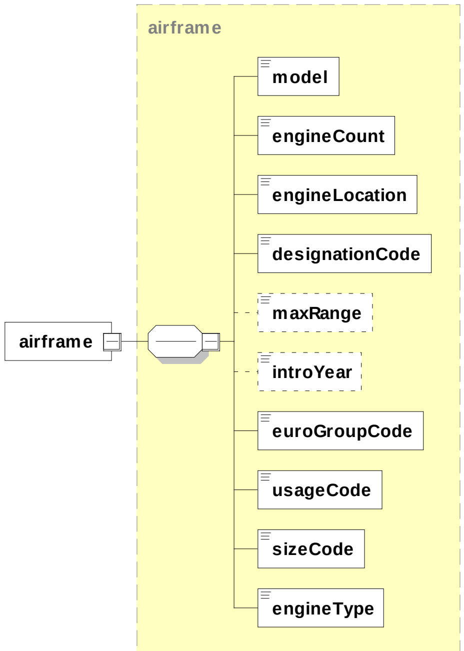
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
anpAircraftId	-	1	a	Record identifier.
airFrameModel	string50	1	a	Air frame model.
engineCode	string50	1		Engine code. Valid values: E (Electric), J (Jet), P (Piston), T (Turboprop).
engineModCode	string50	?		Engine modification code. (AEDT database reference table FLEET.FLT_ENGINE_MODS column ENGINE_MOD_CODE.)

Attributes: None

5.2.2 airframe



Specifies a new airframe.

Structure

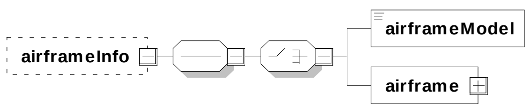
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
model	string50	1	Description of airframe. This must be a new, unique model value.
engineCount	int	?	Number of engines on airframe.
engineLocation	enum	?	Position of engine on airframe. Valid values: F (Fuselage/Tail), W (Wing).
designationCode	string1	?	Type of aviation. Valid values: C (Civil), G (General Aviation), M (Military).
maxRange	int	?	Number of miles airframe can fly fully fueled.
introYear	int	?	Year airframe was introduced.
euroGroupCode	string2	?	European group code for this airframe. Valid values: H1 (Helicopter Light), H2 (Helicopter Heavy), JB (Jet Business), JL (Jet Large), JM (Jet Medium), JR (Jet Regional), JS (Jet Small), PP (Propeller), SS (Supersonic), TP (Turboprop).

XML Tag	Type	Num	Description
usageCode	string1	?	Usage code for this airframe. Valid values: H (Heavy), L (Large), M (Medium), S (Small), T (Light), V (Very Light).
sizeCode	string1	?	Size code for this airframe. Valid values: H (Heavy), L (Large), M (Medium), S (Small), T (Light), V (Very Light).
engineType			Type of engine on this airframe. Valid values: E (Electric), J (Jet), P (Piston), T (Turboprop).

Attributes: None

5.2.3 airframeInfo



Specifies airframe information.

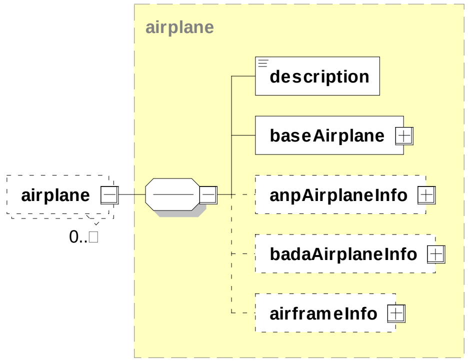
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
airframeModel	string50	1	a	Indicates the airframe model.
airframe	-	1	a	Specifies the airframe. See section 5.2.2.

Attributes: None

5.2.4 airplane



Characterizes an airplane.

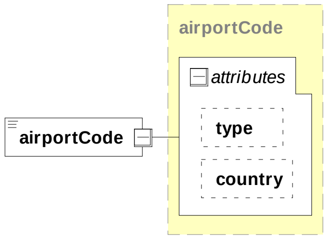
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
description	string255	1	Airplane’s description.
baseAirplane	-	1	Base description of an airplane. See section 5.2.27.
anpAirplaneInfo	-	*	ANP modifications to the airplane’s base description. See section 5.2.12.
badaAirplaneInfo	-	*	BADA modifications to the airplane’s base description. See section 5.2.24.
airframeInfo	-	*	Specifies the airframe model. See section 5.2.3.

Attributes: None

5.2.5 airportCode



Contains an airport's code.

Structure:

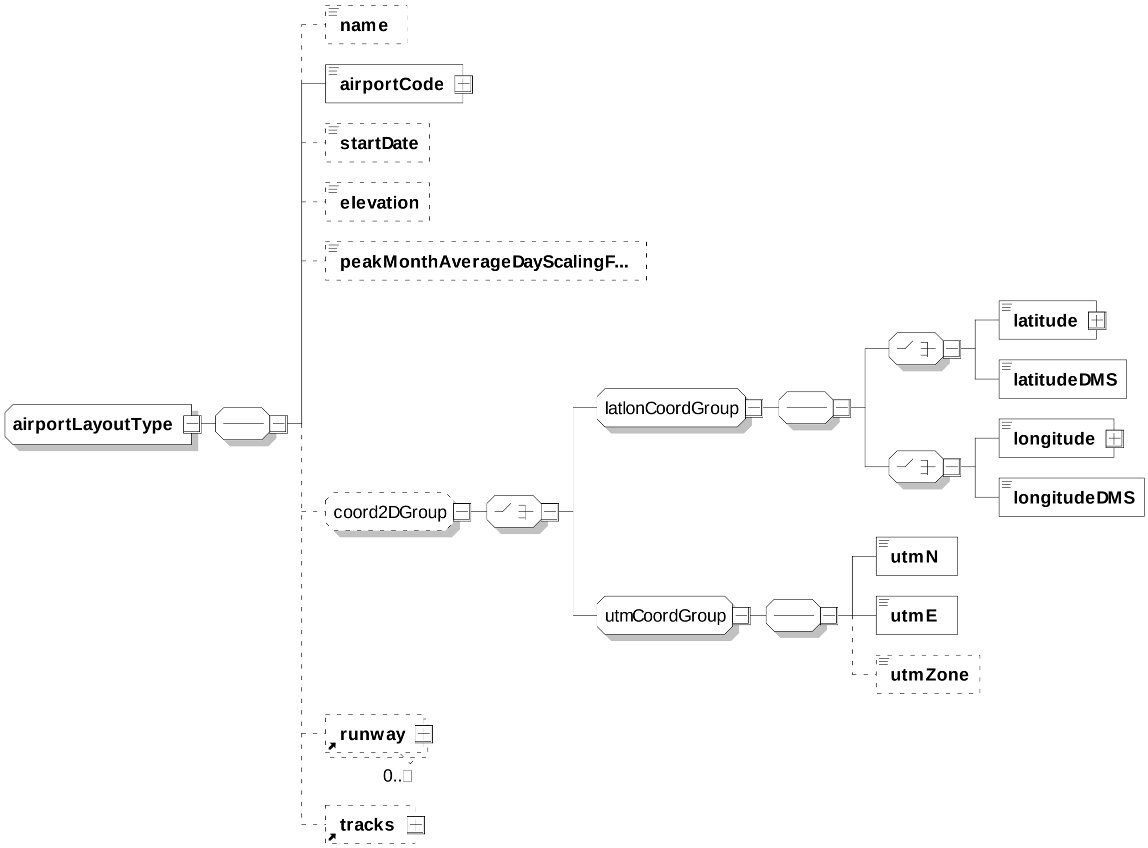
No children elements.

Attributes

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
type	enum	?	Type of airport code. Valid values for new studies: ICAO, IATA, FAA, OTHER. Legacy studies may use ANY and OTHER. Accepted airport codes can be confirmed in the dbo.APT_CODE table in the AEDT Airport database.
country	string3	?	Standard ISO 316601 two-letter country codes. For a list of these codes, see http://unstats.un.org/unsd/methods/m49/m49alpha.htm. Database reference: Airport.APT_CNTRY_CODE.CNTRY_CODE

5.2.6 airportLayoutType



Defines an airport's layout.

Structure

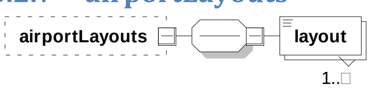
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
name	string255	1	ID of the layout. Must be unique.
airportCode	-	1	ICAO code of airport in the layout. See section 5.2.5.
startDate	date	1	Date airport is included in the study.
elevation	float	?	Elevation of the layout in feet above MSL.
coord2Dgroup	-	1	Type of 2-D coordinates specifying this airport layout. See section 5.3.2.

XML Tag	Type	Num	Description
peakMonthAverageDayScalingFactor	double	1	This scale factor is applied to convert Average Annual Day operations to Peak Month Average Day operations. This is to comply with regulatory reporting requirements for the Peak Month Average Day emissions and fuel burn totals at individual airports.
runway	-	*	Runways included in the layout. See section 5.2.42.
tracks	-	*	Set of flight tracks. See section 5.2.54.

Attributes: None

5.2.7 airportLayouts



Container for one or more airport layouts.

Structure

See section 5.1 for terminology assistance.

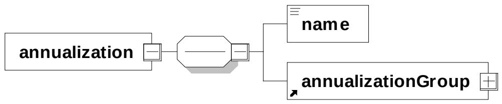
XML Tag	Type	Num	Description
layout	-	?	Describes an airport layout. See section 5.2.6.

Attributes

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
dummy	int	?	Not used.

5.2.8 annualization



Container for one or more annualization groups.

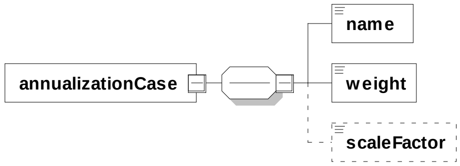
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
name	string255	1	Name of annualization.
annualizationGroup	-	+	Describes an annualization group. See section 5.2.10.

Attributes: None

5.2.9 annualizationCase



An annualization case is a collection of study cases whose results will be weighted in the scenario annualization rollup.

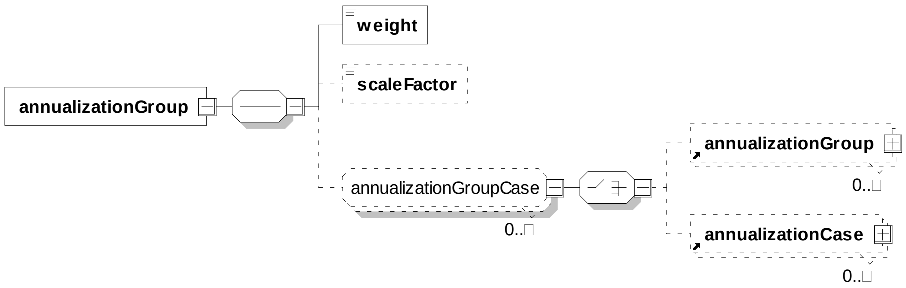
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
name	string255	1	Description of the case.
weight	double	1	Weight associated with the case.
scaleFactor	float	?	Scale factor applied to results for the case.

Attributes: None

5.2.10 annualizationGroup



Contains one or more weighted annualization group cases.

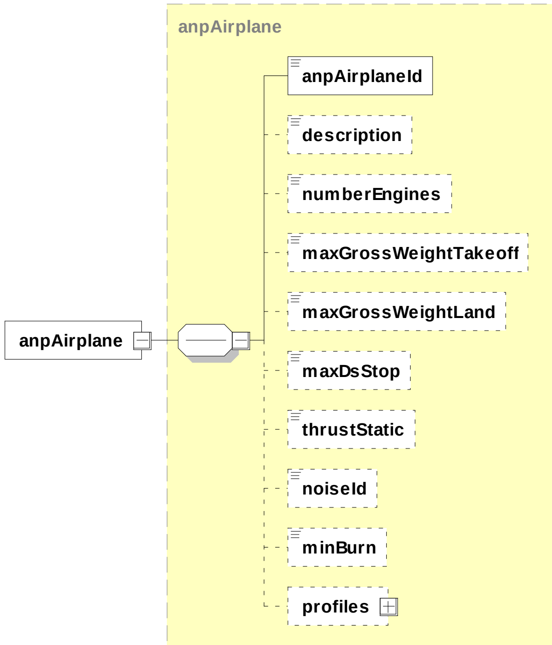
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
Weight	double	1	Weight associated with the annualization group.
scaleFactor	float	?	Scale factor applied to results for the annualization group.
annualizationGroupCase	-	+	A list of annualization groups or cases. See section 5.3.1.

Attributes: None

5.2.11 anpAirplane



Detailed characteristics of a new user-defined ANP airplane.

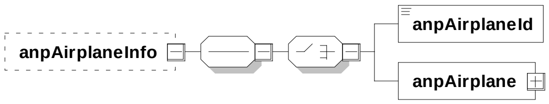
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
anpAirplaneId	string12	1	ID of ANP airplane. Must be a new, unique value.
description	string40	?	Description of ANP airplane.
numberEngines	int	?	Number of engines on this airplane.
maxGrossWeightTakeoff	int	?	Maximum gross weight on takeoff (min = 0, max = 999999, lbs).
maxGrossWeightLand	int	?	Maximum gross weight on landing (min = 0, max = 999999, lbs).
maxDsStop	int	?	FAR landing field length at maximum landing weight (min =0, max = 20000, feet).
thrustStatic	int	?	Static rated thrust or 100% thrust (lb, min =0, max = 200000).
noiseId	string12	?	ID of a Noise Group.
minBurn	double	?	Minimum fuel burn rate. (kg/sec)
profiles	-	?	ANP Profiles associated with this airplane. See section 5.2.21.

Attributes: None

5.2.12 anpAirplaneInfo



Detailed characteristics of an airplane.

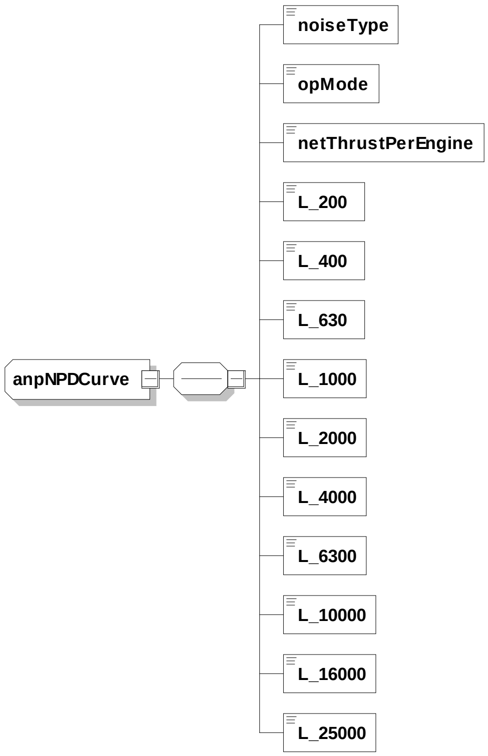
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	choice	Description
anpAirplaneId	string12	1	a	ID of an existing ANP airplane model. Database reference: FLEET.FLT_ANP_AIRPLANES.ACFT_ID
anpAirplane	-	1	a	Description of a new user-defined ANP airplane. See section 5.2.11.

Attributes: None

5.2.13 anpNPDCurve



Creates a new user-defined Noise-Power-Distance (ANP) curve.

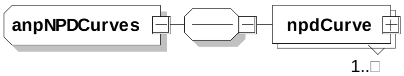
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
noiseType	string12	1	Type of noise described by this curve. Must be a new, unique value.
opMode	string10	?	Operation mode. (A = Approach, D = Depart, X = AfterBurner)
netThrustPerEngine	float	?	Net thrust per engine (min = 0.10, max = 99999.00, lbs. or percentage depending on parent noise group THRUST_SET_TYPE value).
L_200 through L_25000	double	?	Decibel level at specific distances – 200 ft through 25,000 ft.

Attributes: None

5.2.14 anpNPDCurves



Creates a new set of user-defined ANP curves.

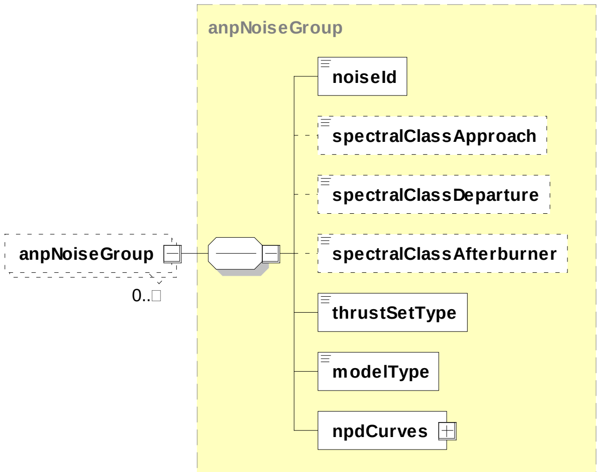
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
npdCurve	-	+	Base noise data interpolated/extrapolated upon according to slant range distance and thrust setting for aircraft. See section 5.2.13.

Attributes: None

5.2.15 anpNoiseGroup



Describes an ANP noise group.

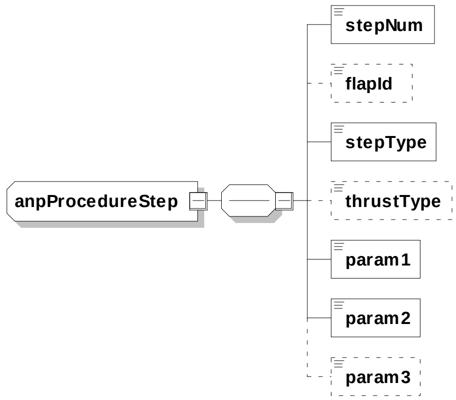
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
noiseId	string12	1	Noise group's ID.
spectralClassApproach	short	?	Spectral class number for approach (min = 0, max = 999).
spectralClassDeparture	short	?	Spectral class number for departure (min = 0, max = 999).
spectralClassAfterburner	short	?	Spectral class number for afterburner (min = 0, max = 999).
thrustSetType	string1	?	Type of thrust setting. Valid values: L (pounds), P (percent), Y (other).
modelType	string1	?	Type of distance-duration model. Valid values: I (INM), N (NoiseMap).
npdCurves	-	?	See section 5.2.14.

Attributes: None

5.2.16 anpProcedureStep



Describes an ANP procedure step.

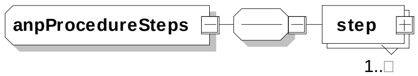
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
stepNum	int	1	Step number of the procedure. Must be unique in a sequence.
flapId	string6	?	Flap-setting identifier. Database reference: STUDY.FLT_ANP_AIRPLANE.FLAPS.FLAP_ID
stepType	string1	?	Type of step.
thrustType	string	?	Type of thrust.
param1 through param3	float	?	Parameters particular for this step type (min = -9999.0, max = 60000.0),

Attributes: None

5.2.17 anpProcedureSteps



Container for a set of ANP procedure steps.

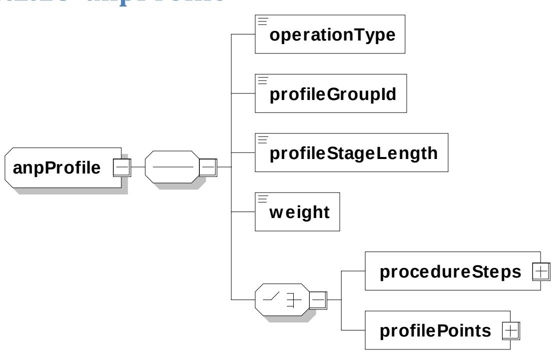
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
step	-	+	An ANP procedure step. See section 5.2.16.

Attributes: None

5.2.18 anpProfile



Describes an ANP profile.

Structure

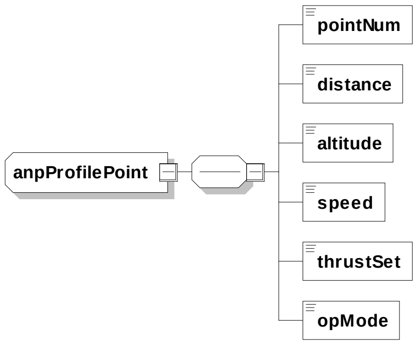
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
operationType	string1	1		Operation associated with this profile. Valid values: A (Approach), D (Depart), T (Touch&Go), F (CircuitFlt), V (OverFlt)
profileGroupId	string8	?		Profile group identifier. Valid values: STANDARD, NOISEMAP (INM standard data).

XML Tag	Type	Num	Choice	Description
profileStageLength	string1	?		Profile stage number (min = 1, max = 9). Approach stage numbers are not related to trip distance. There is only one standard approach profile for most standard aircraft and its stage number is set to 1. Approach stage numbers are used to distinguish members of a group. For example, approach stage can mean different kinds of approaches (e.g. 1 = 3 degree approach, 2 = 5 degree approach).
weight	int			Aircraft weight during this operation type (min = 0, max = 999999, lbs).
procedureSteps	-	?	a	Set of procedure steps associated with this profile. See section 5.2.17.
profilePoints	-	?	a	Set of points associated with this profile. See section 5.2.20.

Attributes: None

5.2.19 anpProfilePoint



Individual point along an ANP profile.

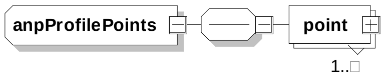
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
pointNum	short	1	Point index number. Must be sequential and unique, starting at 1.
distance	float	1	Distance along the ground relative to start (min = -9999999.9, max = 9999999.9, feet).
altitude	float	1	Altitude of aircraft (min = -9999, max = 60000, feet).
speed	float	1	Ground speed at this point (min = 0, max = 600, knots).
thrustSet	float	?	Corrected net thrust per engine at this point (min = 0.1, max = 99999, klbs or % max thrust).
opMode	string1	?	Operational mode. Valid values: A (Approach), D (Departure), X (Overflight).

Attributes: None

5.2.20 anpProfilePoints



Container of ANP profile points.

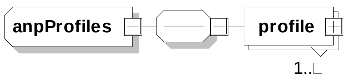
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
point	-	+	An ANP profile point. See section 5.2.19.

Attributes: None

5.2.21 anpProfiles



Container for a set of ANP profiles.

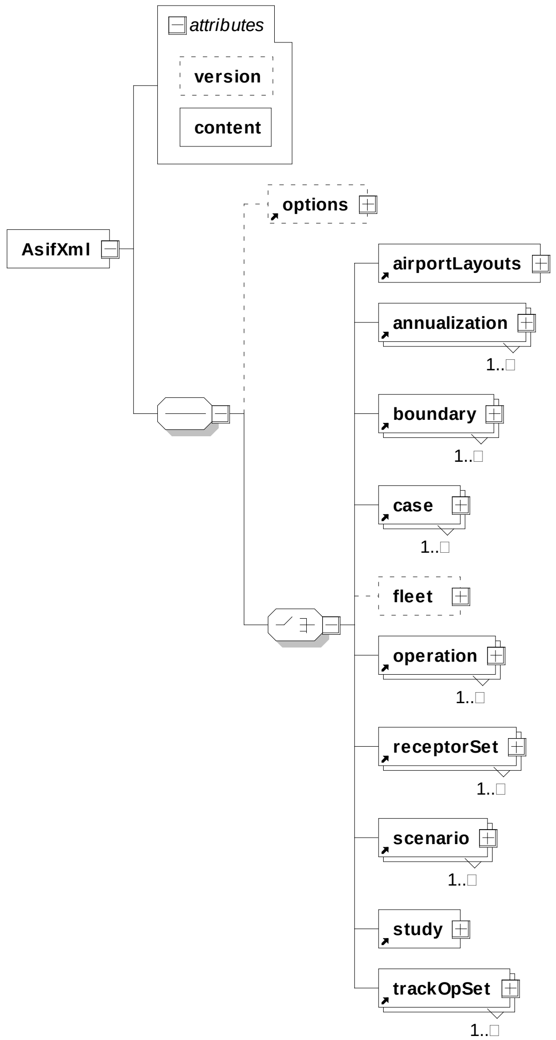
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
profile	-	+	One or more ANP profiles. See section 5.2.18.

Attributes: None

5.2.22 AsifXml



Root node of the ASIF tree.

Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
options	-	?		Contains options applied to the study. See section 5.2.37.
airportLayouts	-	+	a	Contains layouts for ASIF partial import into an existing study. See section 5.2.7.

XML Tag	Type	Num	Choice	Description
annualization	-	*	a	Contains annualizations for ASIF partial import into existing study. See section 5.2.8.
boundary	-	*	a	Contains study area boundaries for ASIF partial import into existing study. See section5.2.28.
case	-	+	a	Contains scenario cases for ASIF partial import into existing study. See section 5.2.29.
fleet	-	?	a	Contains study fleet data for ASIF partial import into existing study.
operation	-	*	a	Contains operations data for ASIF partial import into existing study. 5.2.35
receptorSet	-	*	a	Contains receptor sets for ASIF partial import into existing study. See section 5.2.40.
scenario	-	+	a	Contains scenarios for ASIF partial import into existing study. See section 5.2.44.
study	-	1	a	Contains a full ASIF study. See section 5.2.47.
trackOpSet	-	+	a	Contains trackOpSets for ASIF partial import into existing study. See section 5.2.52,

Attributes

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
version	string16	?	Study version. Used for revision control of a study.
content	enum	1	Describes general content of the study. Valid values: airportLayouts, annualization, boundary, case, fleet, receptorSet, scenario, study, trackOpSet.

5.2.23 badaAirplane

Describes a new user-defined BADA airplane.

Structure

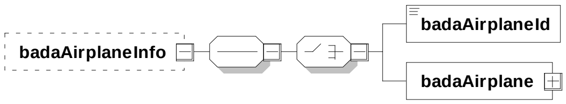
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
badaAirplaneId	string12	1	ID of a BADA airplane model. Must be unique.
mfgDescription	string255	?	Manufacturer description.
referenceAircraftMass	float	?	Minimum aircraft mass (min = 0.0, max = 455.0, metric ton).
minAircraftMass	float	?	Minimum aircraft mass (min = 0.0, max = 455.0, metric ton).

XML Tag	Type	Num	Description
maxAircraftMass	float	?	Maximum aircraft mass (min = 0.0, max = 455.0, metric ton).
maxPayloadMass	float	?	Maximum payload mass (min = 0.0, max = 455.0, (metric ton).
weightGradient	float	?	Weight gradient on maximum altitude (min = 0.0, max = 10.0, feet/kg).
maxOperatingSpeed	float	?	Maximum operating speed (min = 0.0, max = 600.0, knots cas).
maxOperatingMachNumber	float	?	Maximum operating Mach number (min = 0.0, max = 10.0, mach).
maxOperatingAltitude	float	?	Maximum operating altitude (min = -9999.0, max = 60000.0, feel MSL).
maxAltitudeAtMaxTakeoffWeight	float	?	Maximum altitude at maximum takeoff weight and ISA (min = -9999.0, max = 60000.0, feel MSL).
temperatuerGradientOnMaximal Altitude	float	?	Temperature gradient on maximum altitude (min = -1000.0, max = 10.0), feet/degrees C).
wingSurfaceArea	float	?	Wing surface area (min = 0.0, max = 1000.0, square meters).
buffetOnsetLiftCoeff	float	?	Buffet onset lift coefficient (jet only) (min = 0.0, max = 10.0).
buffetGradient	float	?	Buffeting gradient (jet only) (min = 0.0, max = 10.0 (mach-1).
machDragCoeff	float	?	Mach drag coefficient (min = 0.0, max = 10.0).
profiles	-	?	A set of BADA profiles for this airplane. See section 5.2.26.

Attributes: None

5.2.24 badaAirplaneInfo



Specifies the BADA information to be used.

Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	choice	Description
badaAirplaneId	string12	1	a	ID of an BADA airplane model to use for this aircraft.
badaAirplane	-	1	a	Description of a new user-defined BADA airplane. See section 5.2.23.

Attributes: None

5.2.25 badaProfile

Describes a BADA profile.

Structure

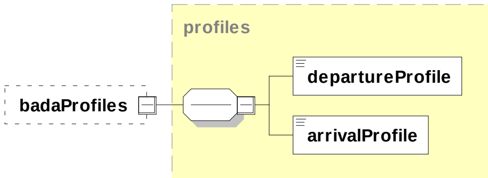
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
massRangeValue	string2	1	Mass range. Valid values: LO (low range), AV (average range), HI (high range).
companyCode1	string3	?	Three-letter company code.
companyCode2	string2	?	Two-letter company code.
companyName	string15	?	Name of airline that uses this procedure.
aircraftVersion	string12	?	Aircraft version to which this procedure applies.
engine	string12	?	Engine identifier.
climbSpeedBelowTransitionAltitud e	short	?	Standard climb speed between 1,500/6,000 and 10,000 feet (min = 0.0, max = 600.0), knots cas).

XML Tag	Type	Num	Description
climbSpeedAboveTransitionAltitude	short	?	Standard climb speed between 10,000 feet and Mach transition altitude (min = 0.0, max = 600.0, knots cas).
climbMachNumber	double	?	Standard climb Mach number above Mach transition altitude (min = 0.0, max = 10.0, mach).
cruiseSpeedBelowTransitionAltitude	short	?	Standard cruise speed between 3,000 and 10,000 feet (min = 0.0, max = 600.0, knots cas).
cruiseSpeedAboveTransitionAltitude	short	?	Standard cruise speed above 10,000 feet until Mach transition altitude (min = 0.0, max = 600.0, knots cas).
cruiseMachNumber	double	?	Standard cruise Mach number above transition altitude (min = 0.0, max = 10.0, mach).
descentMachNumber	double	?	Standard descent Mach number above transition altitude (min = 0.0, max = 10.0, mach).
descentSpeedUnderTransitionAltitude	short	?	Standard descent speed between 3,000/6,000 and 10,000 feet (min = 0.0, max = 600.0, knots cas).
descentSpeedOverTransitionAltitude		?	Standard descent speed above 10,000 feet until Mach transition (min = 0.0, max = 600.0, knots cas).

Attributes: None

5.2.26 badaProfiles



Container for a set of BADA profiles.

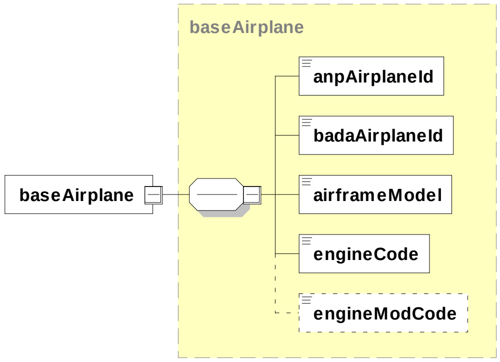
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
profile	-	+	One or more BADA profiles. See section 5.2.25.

Attributes: None

5.2.27 baseAirplane



Base configuration for an airplane.

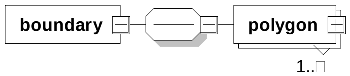
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
anpAirplaneId	string12	?	Airplane's ANP ID. Database reference: STUDY.FLT_ANP_AIRPLANE.ACFT_ID
badaAirplaneId	string10	?	Airplane's BADA ID. Database reference: STUDY_FLT_BADA_AIRPLANE.BADA_ID
airframeModel	string50	?	Airplane's airframe model. Database reference: STUDY.FLT_AIRFRAMES.MODEL
engineCode	string50	?	Airplane's engine code. Database reference: STUDY.FLT_ENGINES.ENGINE_CODE
engineModCode	string50	?	Airplane's engine modification code. Database reference: STUDY.FLT_ENGINE_MODS.ENGINE_MOD_COD E

Attributes: None

5.2.28 boundary



Specifies the boundaries of a study or other element contained within a study.

When a study boundary is specified, all flight paths resulting from departure, arrival, and overflight operations are calculated to and/or from the study boundary.

Structure

See section 5.1 for terminology assistance.

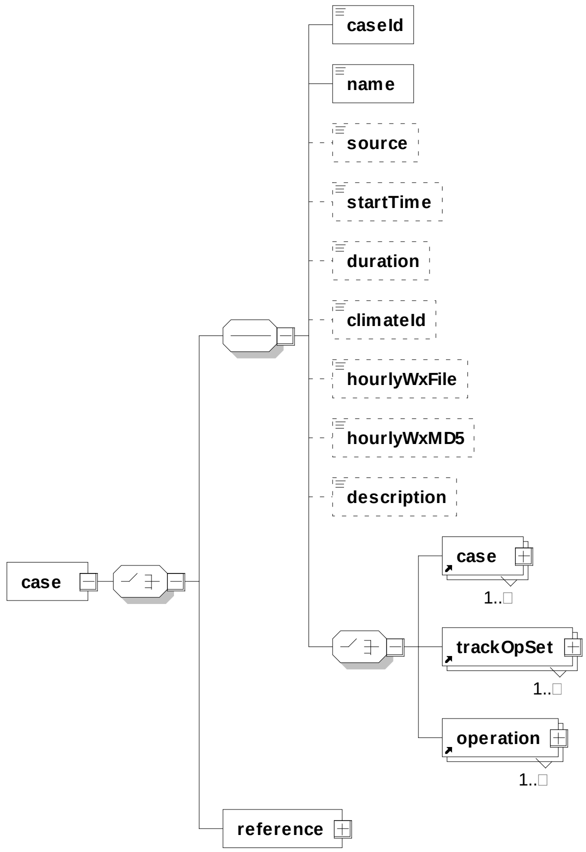
XML Tag	Type	Num	Description
polygon	-	+	Set of coordinates defining the boundary. See section 5.2.38.

Attributes

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
dummy	int	?	Not used.

5.2.29 case



Describes general parameters for a case.

Structure

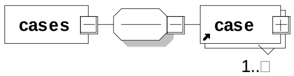
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
caseid	int	1		Case ID.
name	string255	1		The name of the case (must be unique within the scenario).
source	enum	1	a	Emissions source for this case. Valid values: Container, Aircraft, GSE Population, Parking Facilities, Roadways, Stationary Sources, Training Fires.

XML Tag	Type	Num	Choice	Description
startTime	datetime	1		Case's start time. If not defined, the value specified in the scenario element will be used. Must match the value for startTime for the scenario (see section 5.2.44).
duration	int	1		Case's duration. If not defined, the value specified in the scenario element will be used. Must match the value for duration for the scenario (see section 5.2.44). For AEDT 2a this is restricted to 24 hours (1 day). All cases within a scenario must have the same duration as the scenario (hours).
climateId	string8	1		ID of a climate condition. See section 5.2.32.
hourlyWxFile	string255	?		The file containing the hourly weather data used for emissions calculations. This element is not supported in AEDT 2a.
hourlyWxMD5	string16	?		The weather file's MD5 checksum. If not present, the MD5 checksum will be computed for the user at the time of importing the ASIF. This element is not supported in AEDT 2a.
description	string255	?		Description of the case.
case	-		b	Case used in this case.
trackOpSet	-	+	b	Tracks and operations for the case. See section 5.2.52.
operation	-		b	Operation used in this case. See section 5.2.35.
reference	-	S	a	This element is used to reference another case by the scenario and case names. If used, then no other case elements can be used (this element block must be used all by itself inside the case element). See section 5.2.41.

Attributes: None

5.2.30 cases



Placeholder for one or more cases.

Structure

See section 5.1 for terminology assistance.

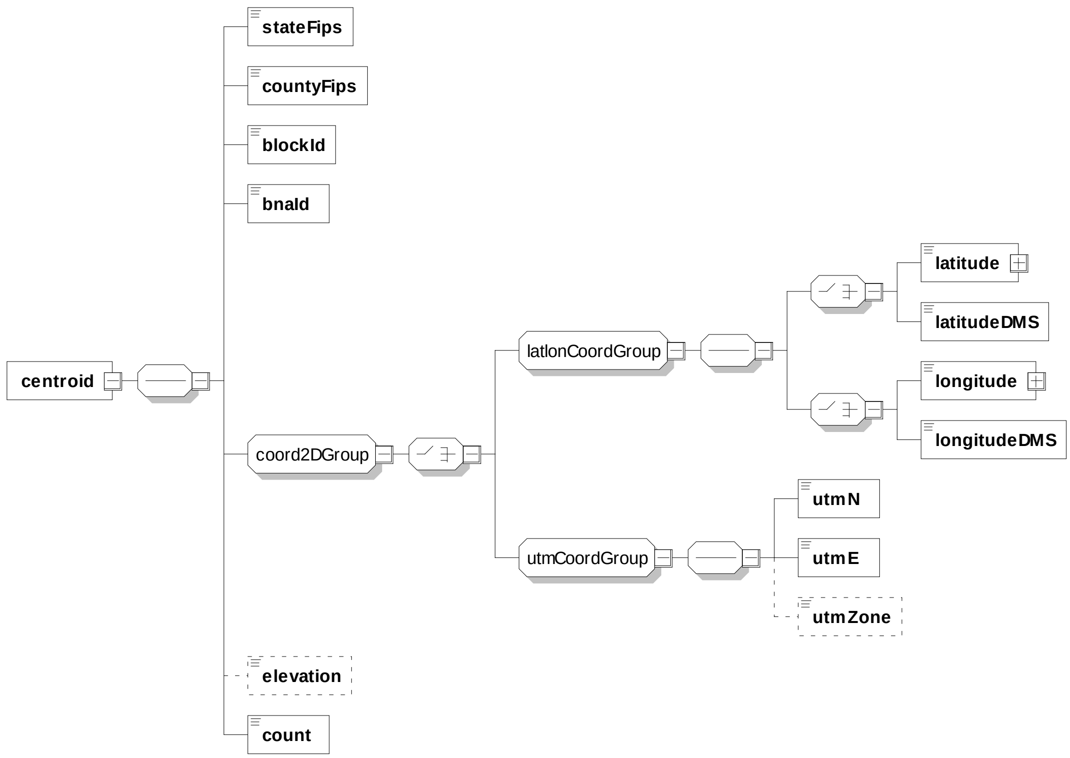
XML Tag	Type	Num	Description
case	-	*	A case Set of coordinates defining the boundary. See section 5.2.29.

Attributes

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
dummy	int	?	Not used.

5.2.31 centroid



Describes the geometric center of a polygon.

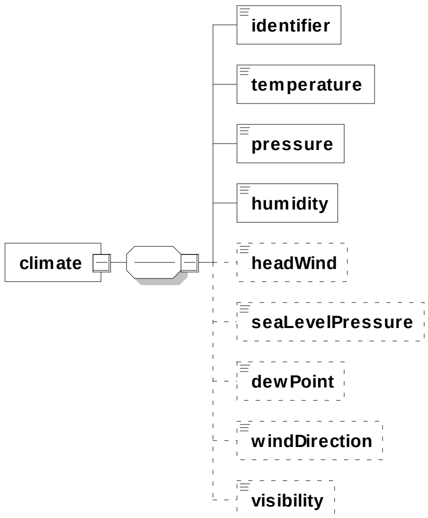
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
stateFips	int	?	Optional census state identifier.
countyFips	int	?	Optional census county identifier.
blockId	int	?	Optional census BLOCK ID.
bnaId	string6	?	Optional census BNA ID.
coord2DGroup	-	1	Type of 2-D coordinates specifying the runway's endpoint. See section 5.3.2.
elevation	float	?	The centroid's elevation above MSL (ft) if terrain not used. If not specified, AEDT 2a will use elevation of operation airport.
count	int	?	The population count of the centroid.

Attributes: None

5.2.32 climate



Characterizes the climate during the study.

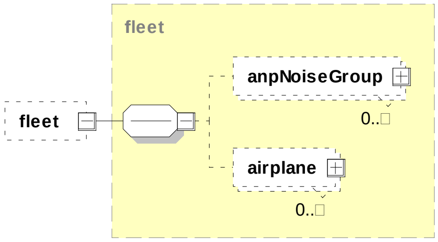
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
identifier	string8	1	Identifier of the climate condition.
temperature	float	?	Temperature in the climate condition. (degress F)
pressure	float	?	Atmospheric pressure in the climate condition (mmHg)
humidity	float	?	Humidity in the climate condition. (mmHg)
headWind	float	?	Velocity of headwind. (knots)
seaLevelPressure	float	?	Atmospheric pressure at sea level. (mmHg)
dewPoint	float	?	Dew point in the climate condition. (degrees F)
windDirection	float	?	Wind direction (degrees)
visibility	float	?	Visibility in the climate condition. (nMi)

Attributes: None

5.2.33 fleet



Describes a fleet of aircraft.

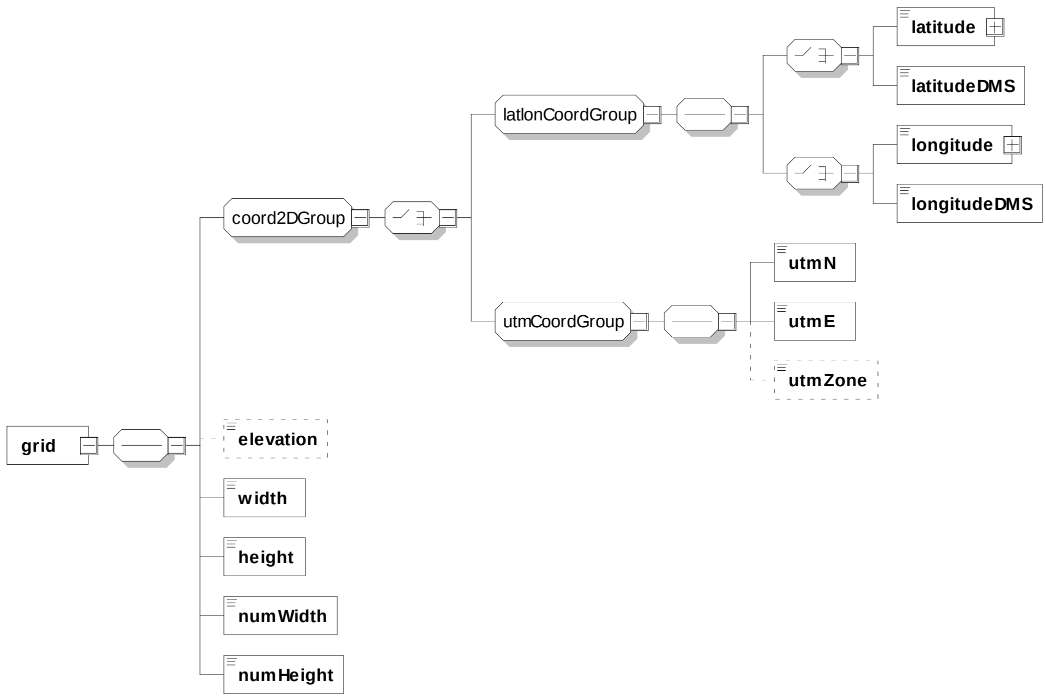
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
anpNoiseGroup	-	*	Contains parameters for Airplane Noise Prediction. See section 5.2.15.
airplane	-	*	Describes an airplane. See section 5.2.4.

Attributes: None

5.2.34 grid



Describes rectangular impact receptor with regularly spaced points.

Structure

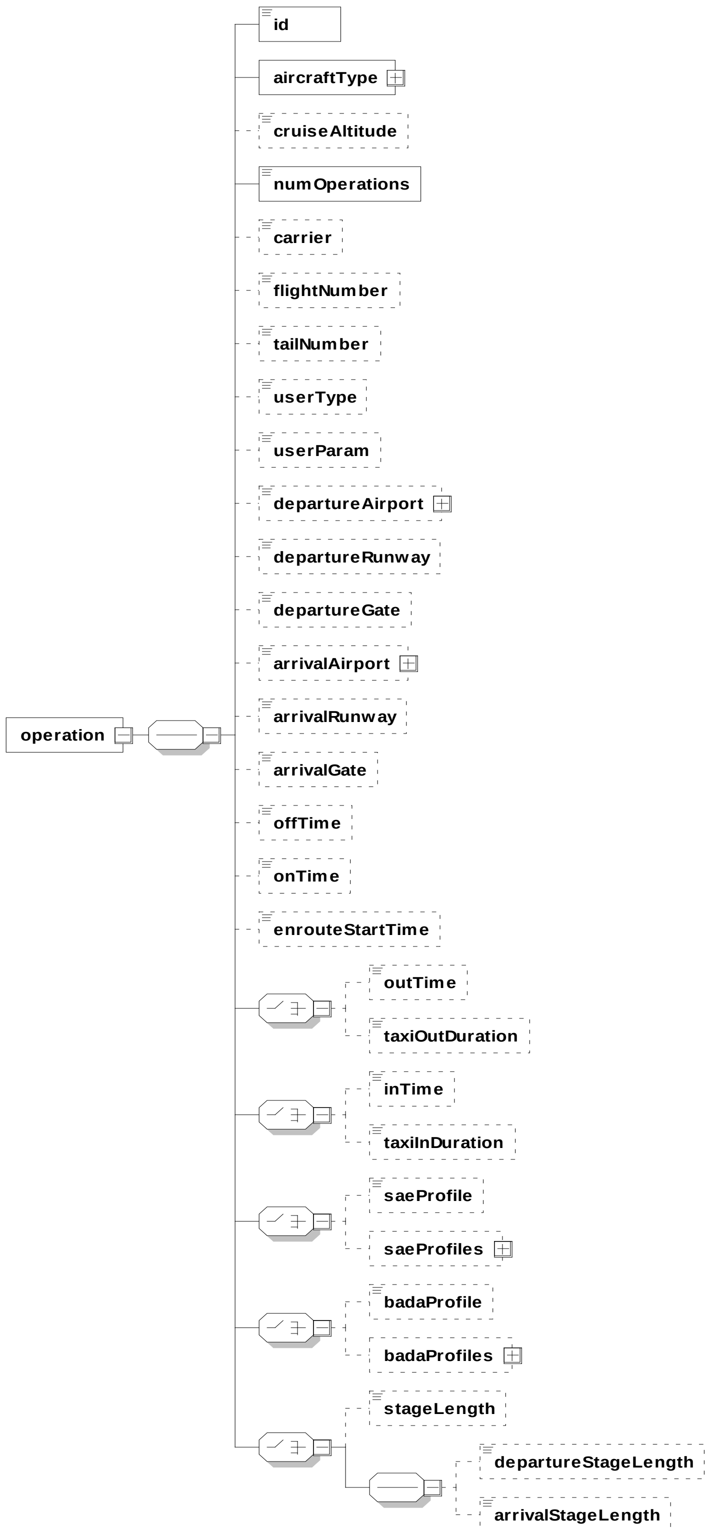
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
coord2DGroup	-	1	Type of 2-D coordinates specifying the grid's origin (southwest corner). See section 5.3.2.
elevation	float	?	The grid's elevation above MSL (ft) if terrain not used. If not specified, AEDT 2a will use elevation of operation airport.
width	float	1	Width of the grid (nmi).
height	float	1	Height of the grid (nmi).
numWidth	int	1	Number of points to spread across the width of the grid. The total number of points in the grid is numWidth × numHeight. Points will be located along width of grid using the formula *i* × (width/numWidth) where *i* is the index of the point (0 … numWidth − 1).

XML Tag	Type	Num	Description
numHeight	int	1	Number of points to spread across the height of the grid. The total number of points in the grid is numWidth × numHeight. Points will be located along height of grid using the formula: *i* × (width/numHeight) where *i* is the index of the point (0 … numHeight − 1).

Attributes: None

5.2.35 operation



Describes an aircraft flight operation.

Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
id	string16	1		User specified identifier for the operation. This field allows the user to create their own identifier for each AirOperation. This field can be unique, degenerate, or even left blank/empty. One purpose served by this field is to allow the user to tie the AEDT AirOperations back to some original data source by setting the id field to an identifying identifier from the original data source. Another purpose could be to set each id to a project specific value for each AirOperation. The id field is used in several AEDT lists and reports that print out the AirOperations. In addition, the Impact Evaluation dialog uses the id as its main method of distinguishing AirOperations when allowing the user to pick and choose operations to be moved to alternative flight tracks. If, however, the user has no outside data sources that need to be tied to the AEDT AirOperations, or if each AirOperation is identical in the sense that no specific AirOperation is more valuable than another or that there will be no intent to distinguish one AirOperation over another, then the suggested approach is to just set the UserID field to unique number or set of characters. This will allow the user to distinguish the AirOperations if the need ever arises. Nevertheless, one can leave all the id fields empty or non-unique set of ids; however, in doing so, the user will be forced to use other identifying fields of the AirOperation if they should ever want to distinguish between AirOperations.
aircraftType	-	1		Type of aircraft in the flight. See section 5.2.1.
cruiseAltitude	double	1		Override aircraft cruise altitude for this operation (ft).
numOperations	double	1		Number of operations comprising this operation.
carrier	string4	?		Carrier flying the flight. Not fully supported in AEDT 2a.
flightNumber	string16	?		Flight number. Not fully supported in AEDT 2a.
tailNumber	string8	?		Flight's tail number. Not fully supported in AEDT 2a.
userType	string12	?		User-defined aircraft type. Cannot be an aircraftType. Not fully supported in AEDT 2a.

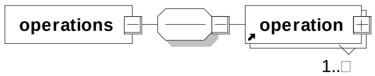
XML Tag	Type	Num	Choice	Description
userParam	string16	?		User-defined parameter associated with the operation. Not fully supported in AEDT 2a.
departureAirport	-	S		Departure airport's ICAO code. Required if the operation is used with a <flight> or <operation> element. Also required if used with a <trackOpSet> modeling departures, circuits, runups, or touch-and-goes. See section 5.2.5.
departureRunway	string8	S		Airport's departure runway ID. Required if the operation is used with a <flight> or a <trackOpSet> modeling departures, circuits, runups, or touch-and-goes.
departureGate	string4	S		Airport's departure gate. Not fully supported in AEDT 2a.
arrivalAirport	-	S		Arrival airport's ICAO code. Required if the operation is used with a <flight> or <operation> element. Also required if used with a <trackOpSet> modeling arrivals, circuits, runups, or touch-and-goes. See section 5.2.5.
arrivalRunway	string8	S		Airport's arrival runway ID. Required if the operation is used with a <flight> or a <trackOpSet> modeling arrivals, circuits, runups, or touch-and-goes.
arrivalGate	string4	S		Airport's arrival gate. Not fully supported in AEDT 2a.
offTime	datetime	S		Wheels-off time. Required for any departure or runup, circuit, runup, or touch-and-go operation.
onTime	datetime	S		Wheels on time. Required for any arrival operation.
enrouteStartTime	datetime	S		Time aircraft reaches the first en route node. Required for en route or overflight flights. Not fully supported in AEDT 2a
outTime	datetime	S	a	Time aircraft pushed back from the gate for a departure. When present, taxiOutDuration = (offTime − outTime). Not fully supported in AEDT 2a.
taxiOutDuration	int	S	a	Number of seconds during taxi-out. Required for emissions modeling, optional for noise modeling. Not fully supported in AEDT 2a.
inTime	datetime	S	b	Time aircraft arrives at arrival gate. When present, taxiInDuration = (onTime – inTime).
taxiInDuration	int	S	b	Number of seconds during taxi-in. Required for emissions modeling, optional for noise modeling.

<table>
<tr><th>XML Tag</th><th>Type</th><th>Num</th><th>Choice</th><th>Description</th></tr>
<tr><td>saeProfile</td><td>string8</td><td>*</td><td>c</td><td>Overrides default profile assignment for a flight's arrival and departure phases using characteristics specified by SAE International. Applicable when the override is unambiguously arrival or departure.</td></tr>
<tr><td>saeProfiles</td><td>-</td><td>*</td><td>c</td><td>Overrides default profile assignment for a flight's arrival and departure phases using characteristics specified by SAE International. Applicable when it is necessary to specify both the arrival and departure profiles. See section 5.2.39.</td></tr>
<tr><td>badaProfile</td><td>string8</td><td>*</td><td>d</td><td>Overrides default profile assignment for a flight's arrival and departure phases using characteristics specified by BADA. Applicable when the override is unambiguously arrival or departure.</td></tr>
<tr><td>badaProfiles</td><td>-</td><td>*</td><td>d</td><td>Overrides default profile assignment for a flight's arrival and departure phases using characteristics specified by BADA. Applicable when it is necessary to specify both the arrival and departure profiles. See section 5.2.39.</td></tr>
<tr><td>stageLength</td><td>string1</td><td>?</td><td>e</td><td>Overrides default departure and arrival stage length values. Applicable when the override is unambiguously arrival or departure. If operation type is Arrival, then AEDT will always use 1 for stage length.</td></tr>
<tr><td>departureStageLength</td><td>string1</td><td>?</td><td rowspan="2">e</td><td>Overrides default departure stage length. Applicable if the phase is a departure phase.</td></tr>
<tr><td>arrivalStageLength</td><td>string1</td><td>?</td><td>Overrides default arrival stage length. Applicable if the phase is an arrival phase. If operation type is Arrival, then AEDT will always use 1 for stage length.</td></tr>
</table>

Stage length "M" does not define a specific trip distance, but rather specifies the maximum weight profile for the given aircraft type. In the absence of any distance information, default cruise altitudes for stage "M" profiles are set based on the maximum N-3
stage length available for the aircraft. Users are advised to assign their own cruise altitude to any operation using stage "M" profiles.

Attributes: None

5.2.36 operations



Contains a list of aircraft flight operations.

Structure

See section 5.1 for terminology assistance.

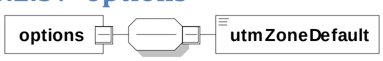
XML Tag	Type	Num	Description
operation	-	+	A flight operation. See section 5.2.35.

Attributes

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
dummy	int	?	Not used.

5.2.37 options



Contains default option values applied to the study.

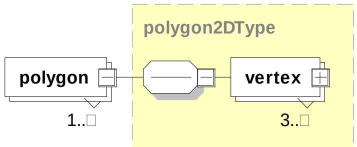
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
utmZoneDefault	int	1	Default UTM zone number.

Attributes: None

5.2.38 polygon



Specifies the shape of a polygon.

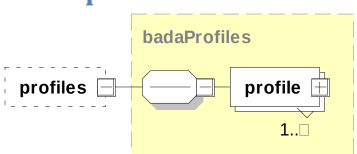
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
vertex	-	3+	A list of vertices defining the polygon. See section 5.2.57.
dummy	int	?	Not used.

Attributes: None

5.2.39 profiles



Contains an arrival and departure profile.

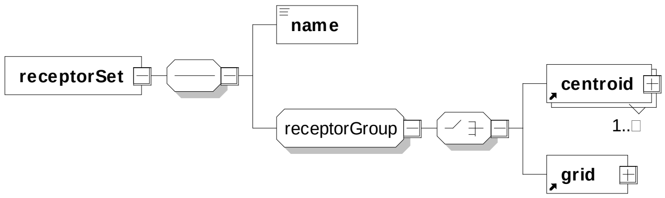
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
departureProfile	string8	1	A flight's departure profile.
arrivalProfile	string8	1	A flight's arrival profile.

Attributes: None

5.2.40 receptorSet



Contains one or more receptor sets at various locations.

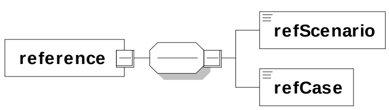
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
name	string255	+	Descriptive name of the receptor set.
receptorGroup	-	+	Description of receptor group at this location. See section 5.3.5.

Attributes: None

5.2.41 reference



Refers to a case by its scenario name and case name.

Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
scenarioRef	string255	1	Name of an existing scenario
caseRef	String255	1	Name of an existing case directly linked to the referenced scenario.

Attributes: None

Conditions required:

- All airport layouts in the referenced scenario must be assigned to the target scenario.
- The referenced case must have a unique name in the new scenario.

5.2.42 runway

Describes dimensions of a runway.

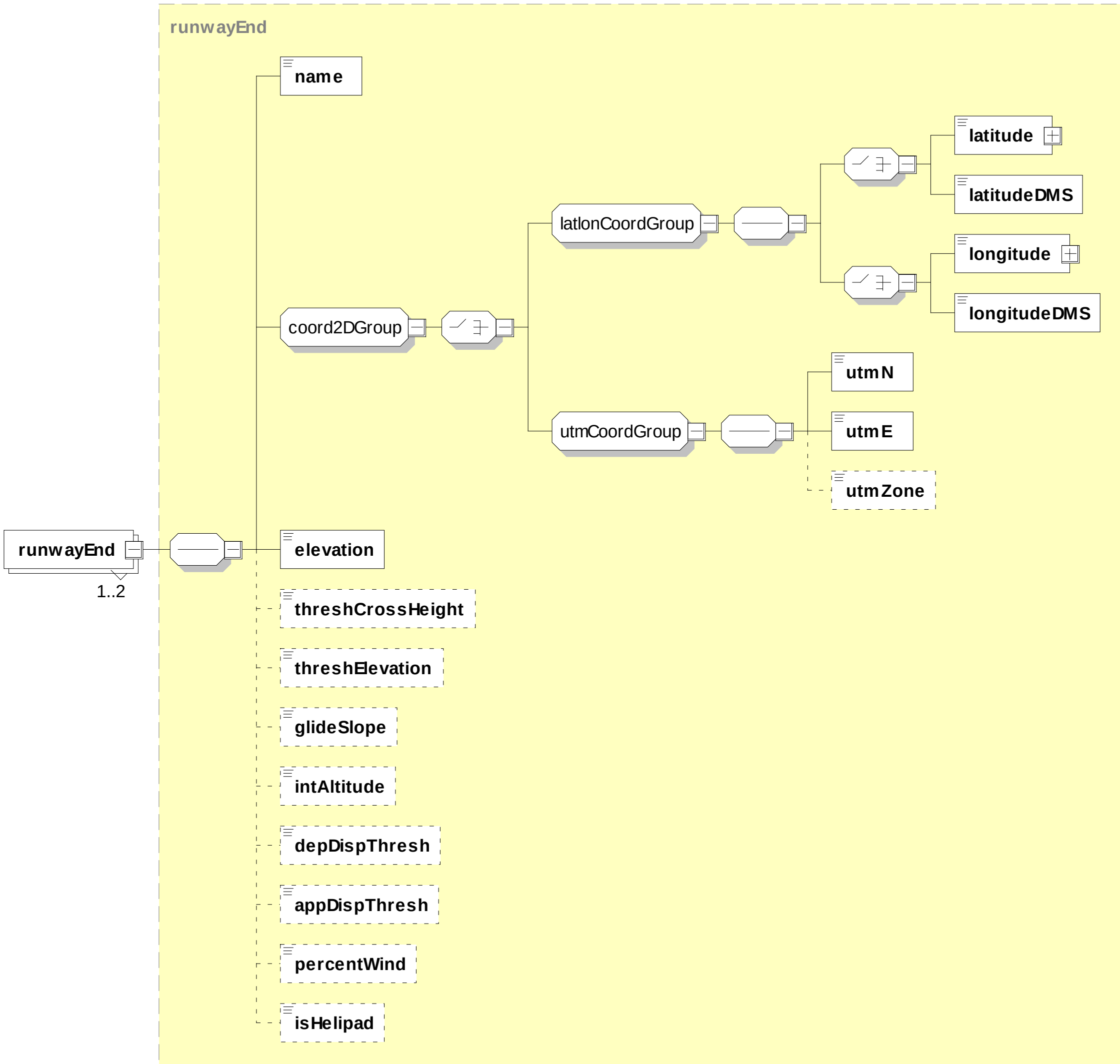
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
length	int	1	Length of runway (feet).
width	int	1	Width of runway (feet).
runwayEnd	-	1-2	Characterizes the runway's endpoint. See section 5.2.43.

Attributes: None

5.2.43 runwayEnd



Characterizes a runway's endpoint.

Structure

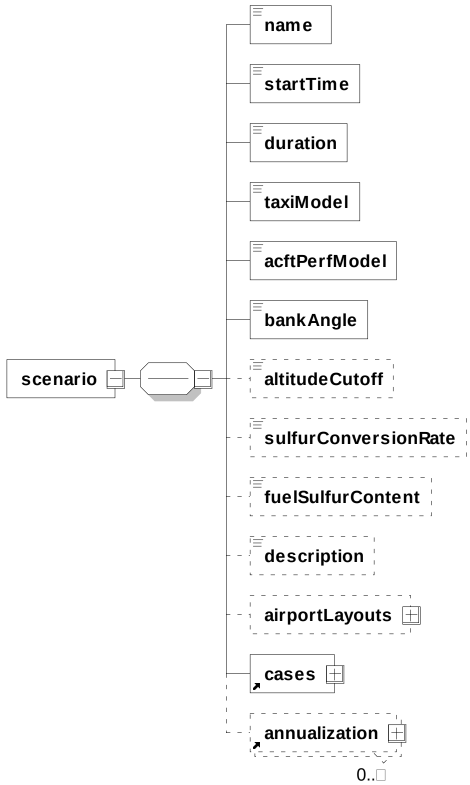
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
name	string8	1	ID of the runway's endpoint.
coord2DGroup	enum	?	Type of 2-D coordinates specifying the runway's endpoint. See section 5.3.2.
elevation	float	?	Runway endpoint's elevation above MSL in feet.
threshCrossHeight	float	?	Approach threshold crossing height AGL (feet).

XML Tag	Type	Num	Description
threshElevation	float	?	Elevation of runway's endpoint (feet MSL).
glideSlope	float	?	Glide slope for runway's endpoint (degrees).
intAltitude	float	?	Runway endpoint's elevation in hundreds of feet.
depDispThresh	float	?	Takeoff displaced threshold (feet).
appDispThresh	float	?	Approach displaced threshold (feet).
percentWind	float	?	Percent change in airport average headwind.
isHelipad	string	?	Indicates if this end of the runway is also a helipad.

Attributes: None

5.2.44 scenario



Describes a flight scenario.

Structure

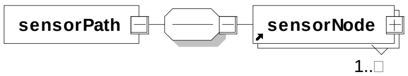
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
name	string255	1	Description of scenario.
startTime	datetime	1	Start time of scenario.
duration	int	1	Scenario's duration (hours).
taxiModel	enum	1	Type of taxi modeling. Valid values: UserSpecified, Sequencing.
acftPerfModel	enum	1	Type of aircraft performance model. Valid values: ICAO, SAE1845.
bankAngle	Boolean	1	Indicates if bank angle calculations should be included in calculations. NOTE: AEDT2A ignores this value and treats all scenarios as if their bank angle value was set to true.

XML Tag	Type	Num	Description
altitudeCutoff	float	1	Altitude to cutoff trajectory modeling for this scenario (nMi MSL). The scenario altitude cutoff only affects noise impact calculation in AEDT 2a. Fuel burn and emissions will be calculated until a flight reaches the study boundary.
sulfurConversionRate	float	1	Portion of sulfur in the fuel that, when combusted, becomes sulfuric acid used for emissions calculations. Default Value: 0.024 (2.4%).
fuelSulfurContent	float	1	Percentage, by weight, of sulfur in the fuel used for emissions calculations. Default Values: 0.0006 (0.06%)
description	string255	?	A description of the scenario.
airportLayouts	-	*	Description of airport layouts used in the scenario. See section 5.2.7.
cases	-	*	The set of cases used in the scenario. See section 5.2.30.
annualization	-	*	Contains annualization for this scenario. See section 5.2.8.

Attributes: None

5.2.45 sensorPath



Describes a flight path based on radar data.

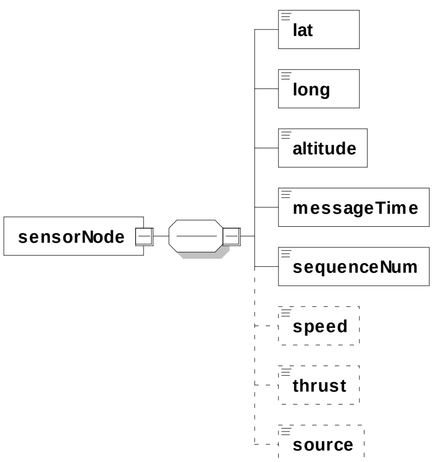
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
sensorNode	-	+	Collection of sensorNodes describing a runway to runway track based on radar data. See section 5.2.46.

Attributes: None

5.2.46 sensorNode



Describes a single node of a radar flight path.

Structure

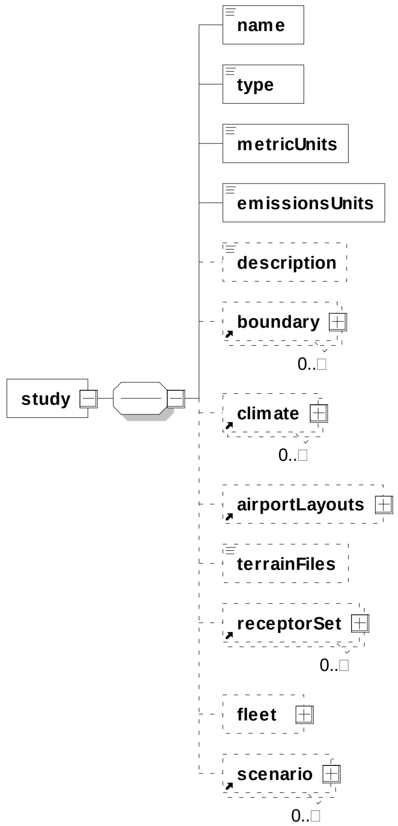
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
lat	double	1	Latitude for this location (decimal degrees).
long	double	1	Longitude for this location (decimal degrees).
altitude	double	1	Altitude at this location (feet)

XML Tag	Type	Num	Description
messageTime	dateTime	1	Time aircraft reaches this location. NOTE: Not used in AEDT 2a.
sequenceNum	int	1	Order of this location in node list.
speed	float	?	Ground speed of aircraft at this location (knots). NOTE: For the terminal-area portion of sensor path results, speeds are based on standard ANP procedures, rather than the speeds given in the sensor path.
thrust	float	?	Thrust of aircraft at this location (lbs). NOTE: Not used in AEDT 2a.
source	string255	?	Source of the data for this node. NOTE: Not used in AEDT 2a.

Attributes: None

5.2.47 study



Contains specific information about a study.

Structure

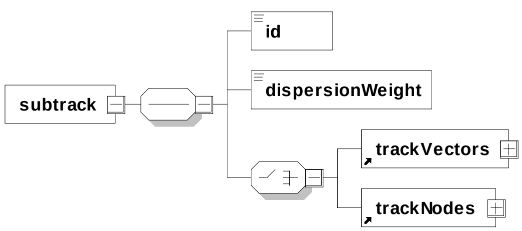
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
name	string255	1	Name of the study.
type	enum	1	Type of study. Valid values: Emissions, Dispersion, Noise and Emissions, Noise and Dispersion. NOTE: AEDT2A only supports the “Noise and Emissions” value.
metricUnits	boolean	1	Indicates if units of measure are metric. TRUE indicates metric units, and FALSE indicates imperial units.
emissionsUnits	enum	1	Emissions units to be used for the study. Valid values: MetricTonnes, Kilograms, Grams, ImperialTons, or Pounds.
description	string255	?	Optional description of the study.

XML Tag	Type	Num	Description
boundary	-	*	Boundaries of areas included in the study. See section5.2.28.
climate	-	*	Prevailing climate during the study. See section 5.2.32.
airportLayouts	-	+	Contains layouts for airports included in the study. See section 5.2.7.
terrainFiles	string255	*	List of files containing descriptions of terrain.
receptorSet	-	*	Define receptor sets for the study. See section 5.2.40.
fleet	-	?	Defines aircraft fleet participating in the study. See section 5.2.33.
scenario	-	+	Defines scenarios for the study. See section 5.2.44.

Attributes: None

5.2.48 subtrack



A collection of segments representing a flight track. AEDT 2a only processes the first subtrack associated with a track.

Structure

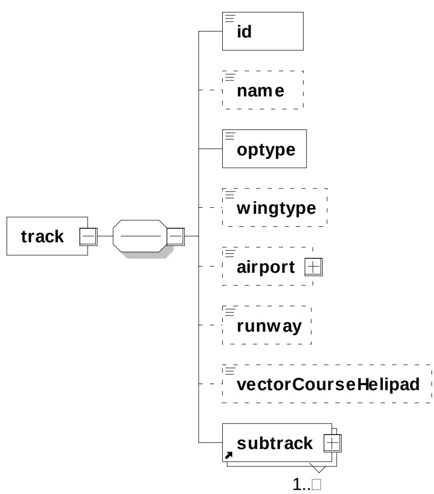
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
id	string16	1		Subtrack ID.

XML Tag	Type	Num	Choice	Description
dispersionWeight	float	1		Dispersion weight associated with this track. All dispersion weights for sibling subtracks must add to one. This field is not used in AEDT 2a, because AEDT 2a only processes the first subtrack associated with a track and applies a value of 1 (100%) to the first subtrack.
trackVectors	-	*	a	A set of vectors describing the subtrack. See section 5.2.56.
trackNodes	-	*	a	A set of nodes describing the subtrack. See section 5.2.51.

Attributes: None

5.2.49 track



A flight's track.

Structure

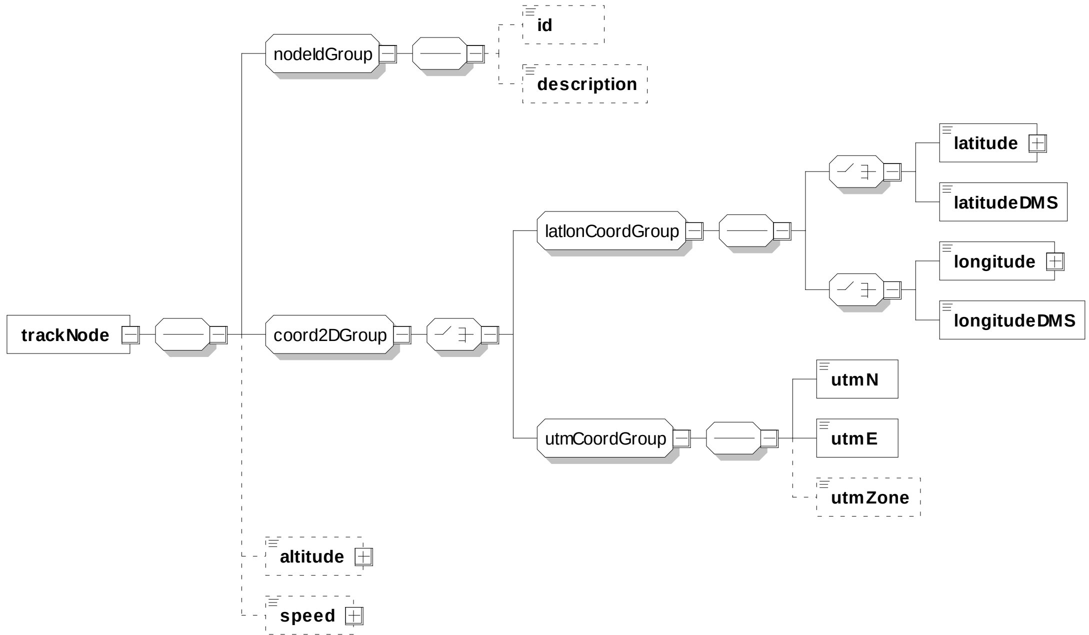
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
id	string16	1	Track ID.
name	string64	?	Descriptive name of the track.

XML Tag	Type	Num	Description
optype	enum	?	Type of operation. Valid values: A, Arrival, D, Departure, V, Overflight, C, Circuit, T, TouchAndGo, R, Runup, W, RunwayToRunway.
wingtype	enum	?	Type of wing. Valid values: F, FixedWing, R, RotaryWing. If not specified, AEDT attempts to determine the wing type based on the optype.
airport	-	?	Airport code associated with the track. See section 5.2.5.
runway	string8	?	Runway identifier.
vectorCourseHelipad	double	?	Initial course heading of helicopter departing from helipad (degrees).
subtrack	-	+	List of subtracks that are part of this track. See section 5.2.48. AEDT 2a only processes the first subtrack associated with a track.

Attributes: None

5.2.50 trackNode



A flight track node.

Structure

See section 5.1 for terminology assistance.

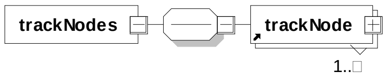
XML Tag	Type	Num	Description
nodeIdGroup	-	*	A list of nodes and their descriptions. See section 5.3.4.
coord2DGroup	-	*	Type of 2D coordinates specifying the node. See section 5.3.2.
altitude	float	*	Node's altitude. Includes attribute node. (nMi MSL)
speed	float	*	Speed of aircraft at node. Includes attribute node. (knots) NOTE: Not used in AEDT 2a.

Attributes: None

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
nodeControlType	enum	*	0, None, 1, AtOrBelow, 2, Match, 3, AtOrAbove.

5.2.51 trackNodes



A set of flight track nodes.

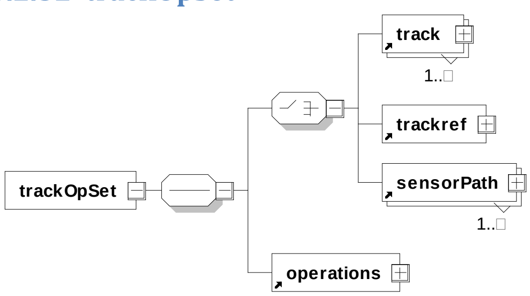
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
trackNode	-	*	A flight track node. See section 5.2.50.

Attributes: None

5.2.52 trackOpSet



Lists tracks and associated operations.

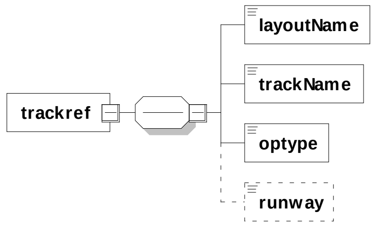
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
track	-	+	a	A flight's track. See section 5.2.49.
trackref	-	+	a	Reference to a flight track. See section 5.2.53.
sensorPath	-	+	a	A complete flight path from runway to runway based on radar data. See section 5.2.45.
operations	-	+		A container listing one or more operations. See section 5.2.36.

Attributes: None

5.2.53 trackref



Reference to a flight track.

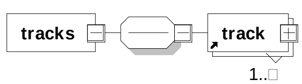
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
layoutName	string255	1	Airport layout associated with this track.
trackName	string64	?	Name of flight track.
optype	enum	?	Type of operation. Valid values: A, Arrival, D, Departure, V, Overflight, C, Circuit, T, TouchAndGo, R, Runup, W, RunwayToRunway.
runway	string8	*	Name of runway on the flight track.

Attributes: None

5.2.54 tracks



A set of flight tracks.

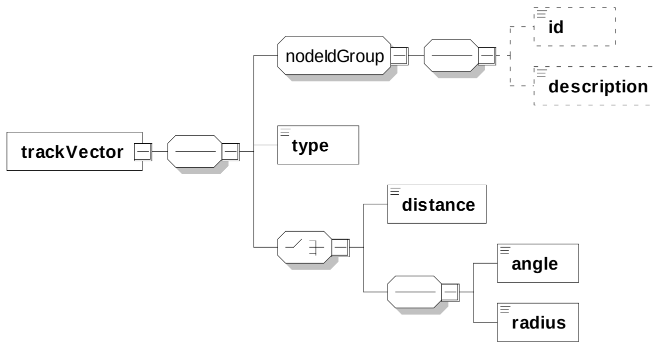
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
track	-	+	A flight track. See section 5.2.49.

Attributes: None

5.2.55 trackVector



A flight track vector.

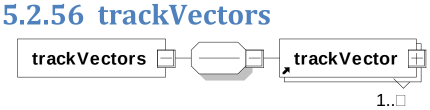
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	choice	Description
nodeIdGroup	-	*		A list of nodes and their descriptions. See section 5.3.4.
type	enum	*		Type of vector. Valid values: S, Straight, L, LeftTurn, R, RightTurn.
distance	float	1	a	Distance flown along this (feet).
angle	float	1	a	Angle of the vector (degrees).
radius	float	1		Radius of the vector (feet).

Attributes: None

5.2.56 trackVectors



A list of flight track vectors.

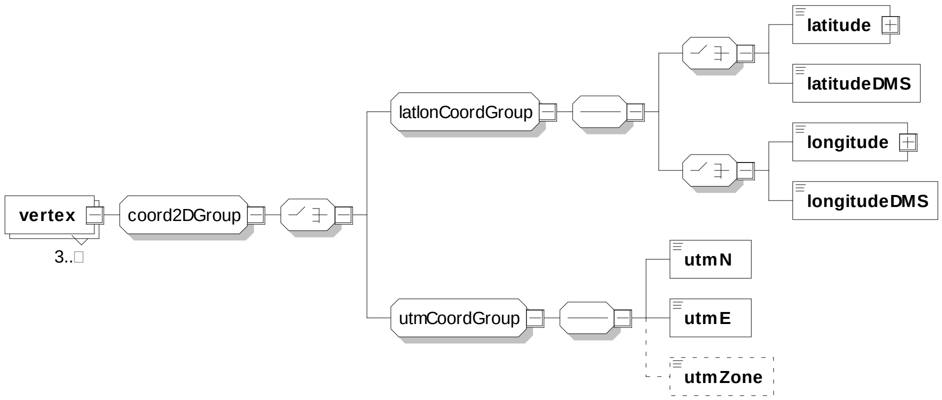
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
trackVector	-	*	A flight track vector. See section 5.2.55.

Attributes: None

5.2.57 vertex



Contains the coordinates of a polygon's vertex.

Structure

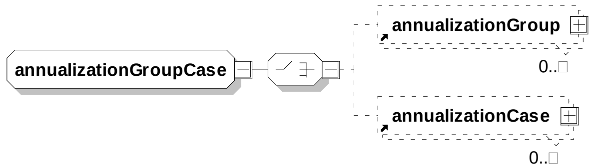
See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
coord2DGroup	enum	1	Type of 2D coordinates specifying the node. See section 5.3.2.

Attributes: None

5.3 Group Descriptions

5.3.1 annualizationGroupCase



A container indicating if a child annualization is done as a group or as a case.

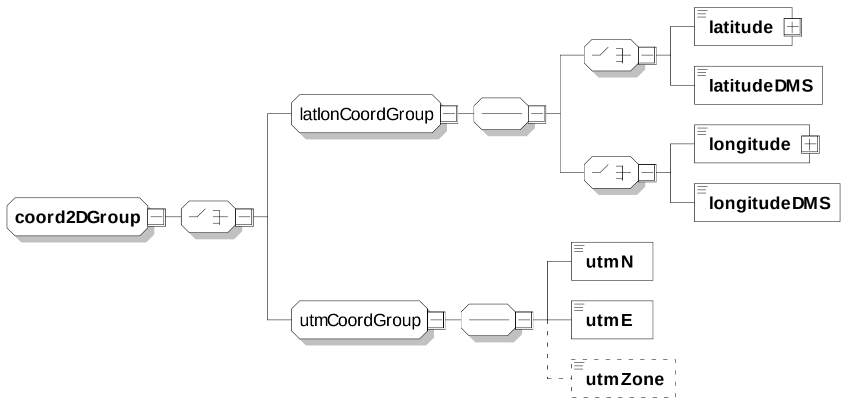
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
annualizationGroup	-	*	a	Indicates child element is annualized as a group. See section 5.2.10.
annualizationCase	-	*	a	Indicates child element is annualized as a case. See section 5.2.9.

Attributes: None

5.3.2 coord2dGroup



Indicates how a 2D coordinate is specified.

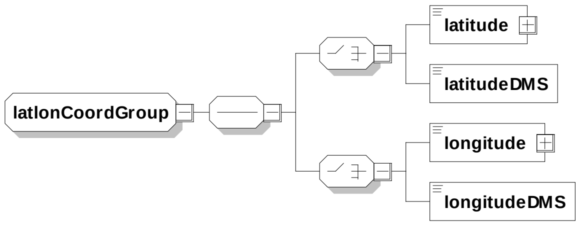
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
latlonCoordGroup	-	1	a	2D coordinate is specified by latitude and longitude. See section 5.3.3.
utmCoordGroup	-	1	a	2D coordinate is specified by latitude and longitude. See section 5.3.6.

Attributes: None

5.3.3 latlonCoordGroup



Specifies a coordinate using latitude and longitude.

Structure

See section 5.1 for terminology assistance.

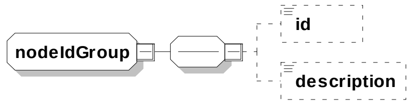
XML Tag	Type	Num	Choice	Description
latitude	-	1	a	Latitude specified as degrees in decimal format. Can include optional attribute positive.
latitudeDMS	-	1	a	Latitude expressed as dd"mm'sss with optional indicator N, n, S, s.
longitude	-	1	b	Longitude specified as degrees in decimal format.
longitudeDMS	-	1	b	Longitude expressed as dd"mm'sss with optional indicator E, e, W, w.

Attributes: None

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
positive	enum	1	For the latitude element, indicates if latitude is north or south of the equator. Valid values: N, n, S, s. For the longitude element, indicates if longitude is east or west of the prime meridian. Valid values: E (default), e, W, w.

5.3.4 nodeIdGroup



Describes a node ID group.

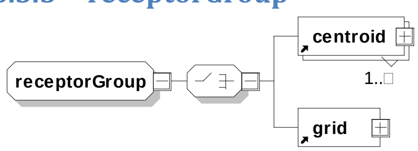
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
id	string16	1	Node group's ID.
description	string16	1	Node group's description.

Attributes: None

5.3.5 receptorGroup



Contains either a centroid or a grid to indicate a location.

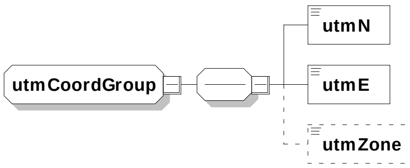
Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Choice	Description
centroid	-	1	a	Describes the location of a centroid. See section 5.2.31.
grid	-	1	a	Describes a grid of points. See section 5.2.34.

Attributes: None

5.3.6 utmCoordGroup



Contains the Universal Transverse Mercator coordinates of a point.

Structure

See section 5.1 for terminology assistance.

XML Tag	Type	Num	Description
utmN	double	1	UTM Northing of the point in decimal meters north of the equator.
utmE	double	1	UTM Easting of the point in decimal meters east from a central meridian.
utmZone	Int	?	UTM Zone of the point. A default zone can be set in the <options> tag.

Attributes: None

6 Procedural Profiles

This section describes procedural profiles for civil aircraft and helicopters. Military aircraft utilize fixed point profiles. For more information on how to setup an anp profile in the ASIF, see section 5.2.18.

6.1 Civil Airplane Procedures

The following sections describe civil aircraft procedure steps and how they are combined into procedural profiles.

6.1.1 Aircraft Profile Operation Types

There are five types of flight operations for aircraft. The valid ASIF identifier is listed in the operation type column of the table.

Operation Type	Full Name	Note
A	Approach	
D	Departure	
T	Touch and go	Touch and go not supported in AEDT 2a
F	Circuit flight	Circuit flight not supported in AEDT 2a
V	Overflight	

6.1.2 Aircraft Procedure Step Types

The procedure step types available in AEDT 2a are listed in the table below. The valid ASIF identifier is listed in the step type column of the table.

Step Type	Full Name	Description
T	Takeoff	Start-roll to takeoff rotation, or touch-and-go power-on point to takeoff rotation
C	Climb	Departure climb to final altitude at constant calibrated airspeed
M	Cruise-Climb	Climb at constant angle to final altitude and speed
A	Accelerate	Departure climb and accelerate to final speed
P	Accel-Percent	Departure climb and accelerate using a constant energy split between acceleration and climbing
V	Level	Maintain altitude and speed
U	Level-Decel	Maintain altitude and reduce speed
W	Level-Idle	Maintain altitude over a given distance with engines at idle
S	Level-Stretch	Special step used to designate where to stretch a circuit flight profile to fit a touch-and-go track

Step Type	Full Name	Description
D	Descend	Descend at constant angle to final altitude
E	Descend- Decel	Descend while reducing airspeed
F	Decend-Idle	Descend at a constant angle with engines at idle
L	Land	Land and roll a given distance
B	Decelerate	Used on approach after touchdown, brake with starting thrust for a given distance

6.1.2.1 Takeoff Step

For a takeoff step, input a flaps identifier and a thrust type. The flaps identifier should not have a U or D prefix because these coefficients were measured on descending flight paths.

MaxTakeoff thrust is typically used for takeoff, but other thrust types are available:

- MaxClimb thrust means that an airplane takes off using reduced thrust, thus requiring a longer runway.
- UserValue thrust means that the user supplies the takeoff thrust value. The thrust value is the corrected net thrust per engine in pounds or in percent of static thrust. AEDT 2a uses the input value at both the start-roll point and at the rotation point.

For MaxTakeoff and MaxClimb thrust, AEDT 2a uses jet or prop coefficients and SAE-AIR-1845 equations to compute thrust values. For jets, the start-roll thrust is computed at 0 knots, and the rotation thrust is computed using the takeoff speed, which comes from another SAE equation. For jets, the thrust is larger at start-roll than at rotation. For props, the thrust is the same at both points and equal to the thrust computed at the rotation point.

6.1.2.2 ClimbStep

For a Climb step, enter a flaps identifier, thrust type, and input the final altitude (the "climb-to" altitude). The final altitude must be higher than the initial altitude. The calibrated air speed on a climb segment is constant, and it is equal to the final speed used on the previous step.

AEDT 2a computes the climb angle and the ground distance based on the airplane weight and average thrust that can be generated for the given conditions. If the computed climb gradient is too small (1%), AEDT 2a processing will stop and log it in the log file.

Typically, MaxTakeoff thrust is used for initial climb segments and MaxClimb thrust for later climb segments, but other thrust types are available:

- UserValue thrust can be assigned to the final climb-to point. AEDT 2a does not adjust this input value for airport elevation, temperature, and pressure.

- UserCutback thrust can be assigned to the whole segment. The difference between UserValue and UserCutback is that AEDT2A applies the user-value-thrust to a point, whereas user-cutback-thrust is applied to a segment. For the cutback case, AEDT2A reduces the thrust over a 1000-foot segment, keeps it constant at the user-cutback value over the climb distance (less 1000 feet), and then returns it to normal thrust over a second 1000-foot segment. The input thrust is corrected net thrust per engine. AEDT 2a does not correct for airport conditions.

6.1.2.3 Accelerate Step

For an Accelerate step, input a flaps identifier, thrust type, climb rate, and final speed (the "accelerate-to" speed). The final speed must be larger than the initial speed.

AEDT 2a uses these input parameters and the SAE-AIR-1845 equations to compute the change in altitude and the distance flown.

The climb rate should be consistent with a sea-level standard-day profile. If necessary, AEDT 2a adjusts the climb rate to account for the actual airport elevation, temperature, and pressure.

Zero climb rate is a valid input. AEDT 2a computes a zero change in altitude, and the thrust is used to accelerate the airplane more quickly.

The five climb thrust types discussed above for the Climb step are also available for an acceleration segment.

6.1.2.4 Accel-Percent Step

For an AccelPercent step, input a flaps identifier, thrust type, energy-share percentage, and final airspeed.

Energy-share comes from the notion that all available thrust is divided between acceleration and climbing. Unlike steps that maintain a constant airspeed while climbing, this step holds the energy-share constant for a given amount of thrust. Enter the percent thrust dedicated to acceleration in the Accelerate Percent box. An input of 70, for example, would result in 70% of thrust going to acceleration and the remaining 100%-70% = 30% of thrust going to climbing.

6.1.2.5 Cruise-Climb Step

For a CruiseClimb step, input a flaps identifier (usually ZERO), final altitude, climb speed, and the climb angle for the segment.

AEDT 2a calculates the distance flown based on the change in altitude and the climb angle. AEDT 2a calculates the corrected net thrust per engine by using the SAE-AIR- 1845 descent equation with a positive angle, rather than a negative angle.

The difference between Climb and CruiseClimb is that thrust for Climb is user defined, whereas AEDT 2a calculates thrust for CruiseClimb based on the input climb angle. Climb thrust is larger than CruiseClimb thrust. Climb steps are used after takeoff when near-maximum thrust is applied. During cruise, less thrust is used in climbing from one altitude to another.

6.1.2.6 Level-Stretch Step

For a LevelStretch step, input a flaps identifier. A Level Stretch step is used to create circuit flight profiles. Its purpose is to define where to put a variable length segment so that a CIR profile fits on top of a TGO track.

- There can be only one Level Stretch step in a CIR profile.
- A Level Stretch step must have a Level step before it and after it. This pair of Level steps should have the same altitude and speed values.

6.1.2.7 Level Step

For a Level step, input a flaps identifier, altitude, speed, and distance flown along the segment. The flaps identifier should be ZERO, or perhaps one with a U prefix (meaning that the landing gear is up).

Input the altitude and speed parameters logically:

- A previous Climb final altitude must equal the Level altitude.
- Also, the Level altitude must equal the next Descend start altitude.
- AEDT 2a computes the amount of thrust needed to maintain level flight at constant speed for the given flaps configuration.

The difference between a Level step and a zero-climb Accelerate step is that the Level step uses a constant speed on the segment, and it uses a smaller value of thrust (and thus, lower noise level) than the Accelerate step. If speed changes during level flight, use a zero-climb Accelerate step.

6.1.2.8 Level-Decel Step

For a Level-Decel step, select input a flaps identifier, altitude, initial airspeed, and distance flown along the segment. Unlike the Level step, airspeed is not held constant but allowed to decrease over the segment. AEDT 2a computes the amount of thrust needed to maintain level flight while decelerating.

The Level-Decel step is subject to the same airspeed and altitude considerations as the Level step, e.g. a preceding climb segment has to end at the same altitude as the Level-Decel step.

6.1.2.9 Level-Idle Step

For a Level-Idle step, input the altitude, initial airspeed, and distance flown along the segment. Airspeed is allowed to decrease over the segment. Unlike Level and Level-Decel steps, thrust is calculated using idle thrust coefficients rather than a force balance.

The Level-Idle step is subject to the same airspeed and altitude considerations as the Level step, e.g. a preceding climb segment has to end at the same altitude as the Level-Idle step.

6.1.2.10 Descend Step

For a Descend step, input a flaps identifier, the starting altitude, starting speed, and the descent angle for the segment.

If a Level or Descend step follows the Descend step, it must have a lower altitude. The following step can have the same or a different speed.

6.1.2.11 Descend-Decel Step

For a DescendDecel step, input a flaps identifier,the starting altitude, starting speed, and the descent angle for the segment. The DescendDecel step differs from the Descend step in that it more explicitly accounts for deceleration effects during thrust calculations.

If a Level or Descend step follows the DescendDecel step, it must have a lower altitude. The following step can have the same or a different speed.

6.1.2.12 Descend-Idle Step

For a DescendIdle step, input the initial airspeed, initial altitude, and descent angle. The DescendIdle step does not require that a flap setting be specified. The other Descend steps that require flap settings utilize a force balance equation to calculate thrust, but this step calculates the aircraft idle thrust directly from engine idle thrust coefficients

6.1.2.13 Land Step

For the Land step, select a flaps identifier and input the touchdown rolling distance, which is the distance that the airplane moves before reversing thrust and/or braking.

The last Descend step and the Land step must both use a flaps identifier that has a D prefix (meaning that the landing gear is down).

AEDT 2a computes the touchdown speed by using a SAE-AIR-1845 equation.

6.1.2.14 Decelerate Step

For a Decelerate step, input the segment distance, the starting speed, and the percent of static thrust at the start of the segment. When applicable, the percent of static thrust at the start of the segment represents the level of reverse thrust.

AEDT 2a uses the percent value and the airplane static thrust to compute a thrust setting value for accessing the NPD curves. For those airplanes that use percent type noise, the percent value is used to access the NPD curves.

6.1.3 Aircraft Thrust Types and Parameters

The thrust types available in AEDT 2a are listed in the table below. The valid ASIF identifier is listed in the thrust type column of the table.

Thrust Type	Full Name
T	MaxTakeoff
C	MaxClimb
N	MaxContinuous
H	ReduceTakeoff
Q	ReduceClimb
S	MaxTakeoffHiTemp
B	MaxClimbHiTemp
M	MaxContinuousHiTemp
G	ReduceClimbHiTemp
P	ReduceClimbHiTemp
I	IdleApproach
J	IdleApproachHiTemp
R	MinimumThrust
K	UserCutback
U	UserValue

The following table shows the remaining parameters needed to create a procedural profile. These fields are called PARAM1, PARAM2, and PARAM3. They take on a different meaning for each combination of operation type, procedure type, and thrust type, see the two tables below.

PARAM	Full Name
THR	Thrust (lbs)
ALT	Altitude (ft AFE)
SPD	Speed (kts)
DIST	Distance (ft)
ANG	Angle (deg)
PCT	Percent
CLM	ClimbRate (ft/min)

Op Type	Step Type	Flap ID	Thrust Type	PARAM1	PARAM2	PARAM3
A,D,T,F,V	V	ID		ALT	SPD	DIST
A,T,F,V	D	ID		ALT	SPD	ANG
A,T,F,	L	ID	T,C,H,Q	DIST	0	0
A,F	B		U	DIST	SPD	PCT
D,F	T	ID	T,C,H,Q	0	0	0
D,F	T	ID	U	0	0	THR
T	T	ID	T,C,H,Q,R	0	SPD	0
T	T	ID	K,U	0	SPD	THR
D,T,F	C	ID	T,C,H,Q,R	ALT	0	0
D,T,F	C	ID	K,U	ALT	0	THR
D,T,F	A	ID		CLM	SPD	0
D,T,F	A	ID		CLM	SPD	THR
A,D,F,V	M	ID		ALT	SPD	ANG
F	S	ID		0	0	0
A	U	ID		ALT	SPD	DIST
A	W			ALT	SPD	DIST
A	E	ID		ALT	SPD	ANG
A	F			ALT	SPD	ANG
D,T,F	P		T,C,H,Q,R	PCT	SPD	0
D,T,F	P		K,U	PCT	SPD	THR

6.1.4 How to Build an Approach Profile

Standard approach procedures generally have four Descend steps, a Land step, and two Decelerate steps, as follows:

- The four Descend steps start at 6000, 3000, 1500, and 1000 feet AFE. They bring an airplane from zero-flaps configuration, terminal-area entrance speed, down to landing-gear/flaps configuration, final-approach speed.
- For most AEDT 2a airplanes, a 3-degree descent angle is used to model IFR approaches. For single-engine piston airplanes and for BEC58P, a 5- degree descent angle is used to model VFR approaches.
- For the Land step, the touchdown-roll distance is 10% of the total rollout distance. For those airplanes using 3-degree approaches, the relationship between the total roll-out distance and the input parameter in the *Airplane Data* window is:

- (Roll-out distance) = 0.9 (Max landing distance)-954
- For those airplanes using 5-degree approaches, the 954-foot value is replaced with 572 feet (the angle is steeper, so the in-air portion of the flight path after crossing the end of the runway is shorter).
- The first Decelerate distance is 90% of the total roll-out distance. The starting speed is less than the touchdown speed. The starting percentage thrust is 40% for narrow-body jets, 10% for wide-body jets, and 40% for props. The first deceleration segment represents reverse thrust action.
- The second Decelerate distance is zero, indicating the end of the profile. The starting speed is 30 knots, representing taxi speed. The starting percentage thrust is 10% of static thrust, representing taxi thrust.

6.1.5 How to Build a Departure Profile

AEDT 2a standard departure procedures for civil jet airplanes tend to follow a pattern (but there are exceptions). A typical civil jet departure profile consists of the following procedure steps:

1. Takeoff using MaxTakeoff thrust and extended flaps.
2. Climb to 1000 feet using MaxTakeoff thrust and takeoff flaps.
3. Accelerate 10-20 knots using MaxTakeoff thrust, takeoff flaps, and 2/3 of the initial climb rate.
4. Accelerate 15-30 knots using MaxTakeoff thrust, reduced flaps, and ½ of the initial climb rate.
5. Accelerate to Vzf (zero-flaps minimum safe maneuvering speed) using MaxClimb thrust, minimal flaps, and 1000-fpm climb rate.
6. Climb to 3000 feet using MaxClimb thrust and zero flaps.
7. Accelerate to 250 knots using MaxClimb thrust, zero flaps, and 1000-fpm climb rate.
8. Climb to 5500 feet using MaxClimb thrust and zero flaps.
9. Climb to 7500 feet using MaxClimb thrust and zero flaps.
10. Climb to 10000 feet using MaxClimb thrust and zero flaps.

A standard departure profile for propeller-driven civil airplanes also tends to follow a pattern of procedure steps:

1. Takeoff using MaxTakeoff thrust and takeoff flaps.
2. Accelerate 10-15 knots using MaxTakeoff thrust, takeoff flaps, and a standard rate of climb.
3. Climb to 1000 feet using MaxTakeoff thrust and takeoff flaps.

4. Accelerate to Vzf using MaxTakeoff thrust, takeoff flaps, and a standard climb rate.

5. Climb to 3000 feet using MaxClimb thrust and zero flaps.

6. Climb to 5500 feet using MaxClimb thrust and zero flaps.

7. Climb to 7500 feet using MaxClimb thrust and zero flaps.

8. Climb to 10000 feet using MaxClimb thrust and zero flaps.

An AEDT 2a standard airplane usually has more than one departure profile. AEDT 2a profiles are distinguished by profile stage numbers from 1 to 9. Departure procedure steps are almost the same for all profile stages. Usually, the change is in the Accelerate step where the final speed value increases for heavier airplanes and the climb rate decreases for heavier airplanes.

6.1.6 How to Build and Overflight Profile

An overflight profile can be built with one procedure step. For example: Level using ZERO flaps, at 5000-foot altitude, at 250 knots, for a distance of 300,000 feet (about 50 nmi).

6.1.7 Airplane Procedure Step Transitions

Procedure steps are combined in prescribed sequences. Certain sequences are not allowed. For example, a climb step may not be followed by a descend step. Procedures must comply with the step transition diagrams provided here.

The step transition diagrams use a simple convention to represent procedures:

- Ellipses represent procedure steps.
- Arrows represent a valid transition from one step to another.
- Arrows point in the direction of the allowed transition – e.g. Land to Decelerate is accepted, but Decelerate to Land is not.
- A double sided arrow means that the transition is valid in both directions.
- An arrow looping back to a step indicates that the step can be repeated.
- A box surrounding two or more steps is used to simplify the diagram.
- Arrows connected to the box apply to each step within.
- Each step within the box can transition to any other within the box.

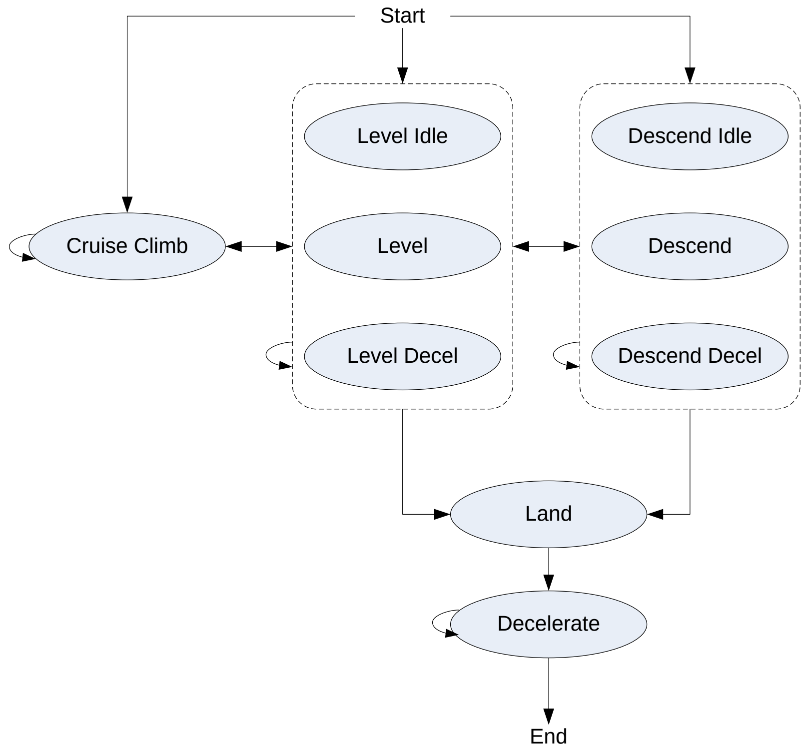


Figure 1: Airplane Approach Step Transition Diagram

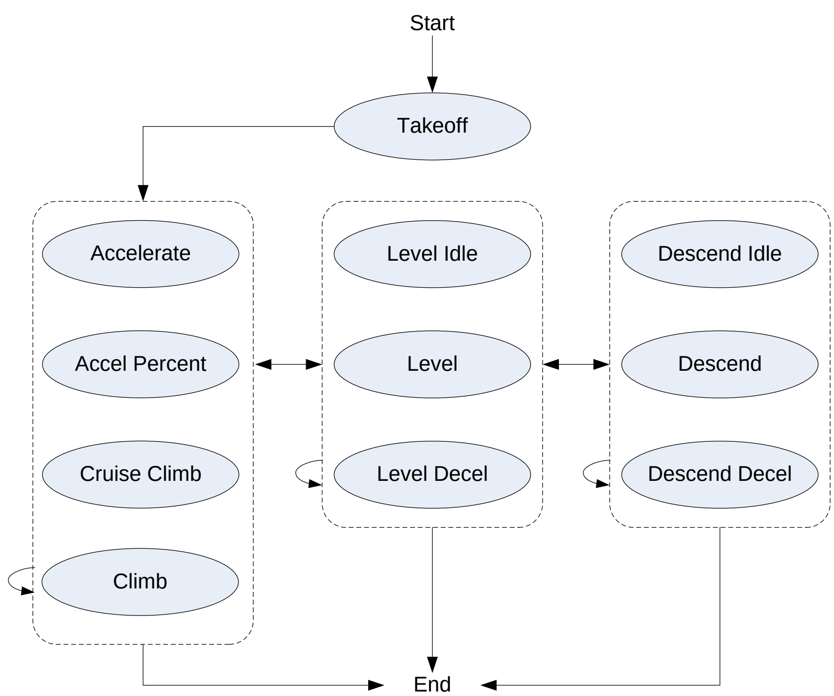


Figure 2: Airplane Departure Step Transition Diagram

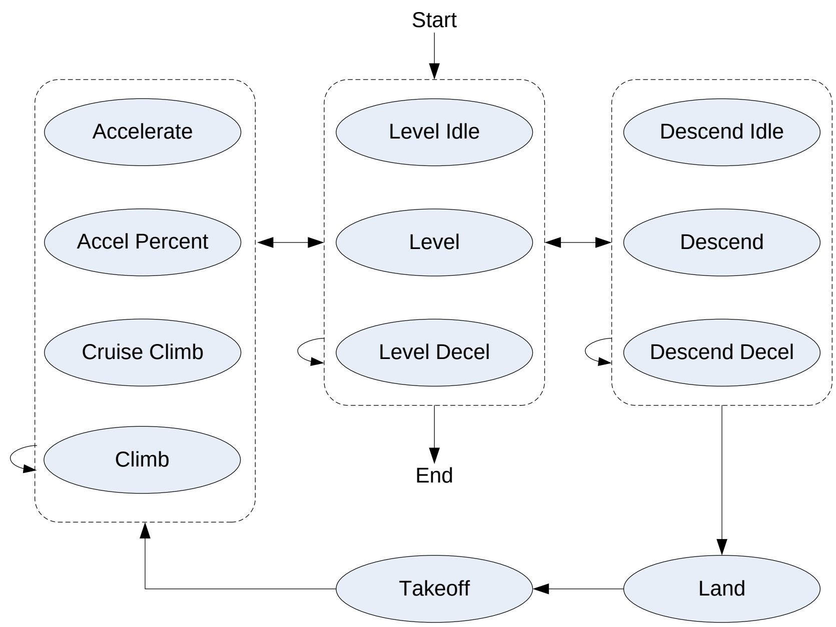


Figure 3: Airplane Touch and Go Step Transition Diagram (not supported in AEDT 2a)

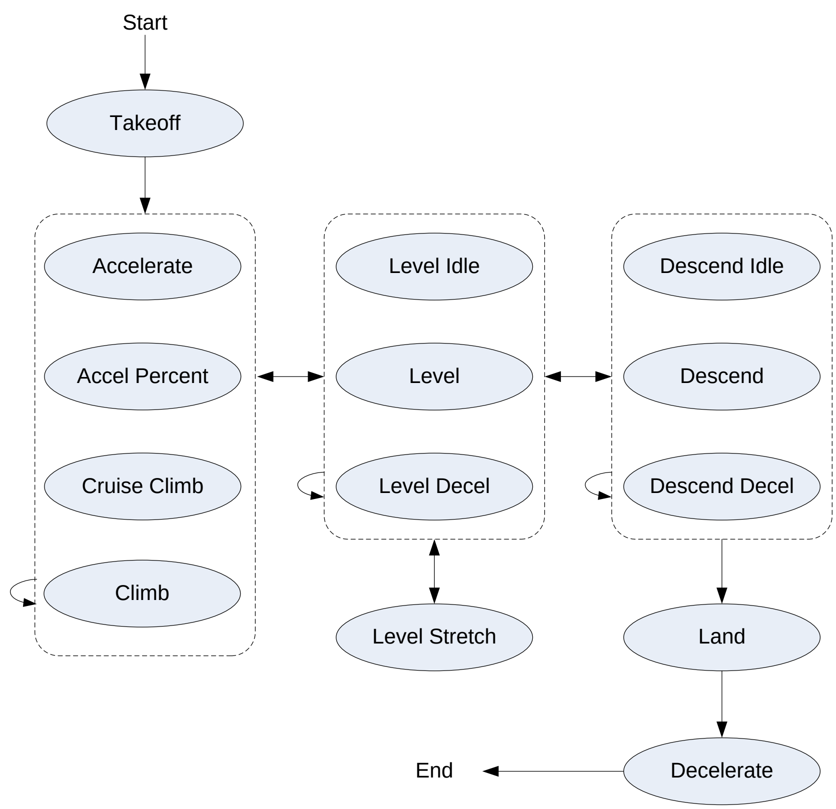


Figure 4: Airplane Circuit Step Transition Diagram (not supported in AEDT 2a)

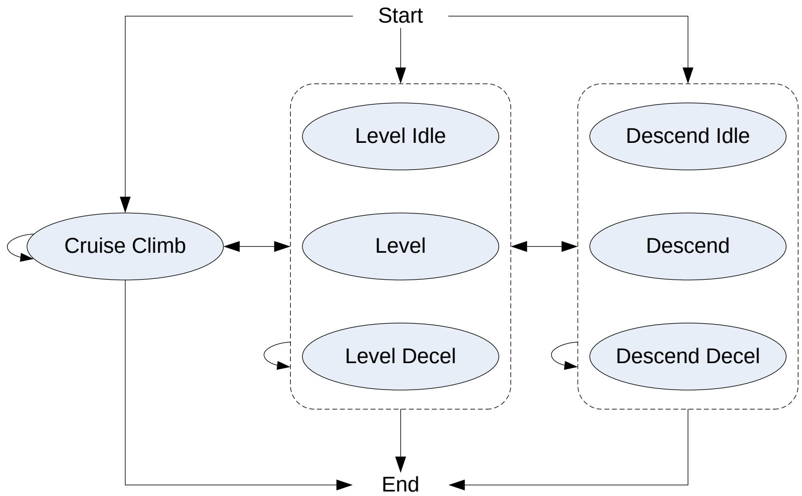


Figure 5: Airplane Overflight Step Transition Diagram

6.2 Helicopter Procedures

The following sections describe helicopter procedure steps and how they are combined into procedural profiles.

Modeling helicopter operations with user defined profiles is not supported in AEDT 2a. W-2

6.2.1 Helicopter Profile Operation Types

There are four types of flight operations for helicopters:

Abbreviation	Single-Letter Identifier	Description	Begin	End	Note
APP	A	Approach	In air	Helipad	
DEP	D	Departure	Helipad	In air	
TAX	T	Taxi	Helipad	Helipad	Taxi not supported in AEDT 2a
OVF	V	Overflight	In Air	In air	

6.2.2 Helicopter Procedure Step Types

The following table describes the procedure steps that are used to define helicopter profiles. The first seven types are the primary NPD operating modes. The next nine are secondary NPD operating modes which can be derived from the primary modes or defined as separate curves. The last step (start altitude) facilitates profile creating as is not associated with an NPD operating mode.

Step Type	Description	State	Parameters
A	Approach at constant speed	Move	Dist Alt
D	Depart at constant speed	Move	Dist Alt

Step Type	Description	State	Parameters
X	Level flyover at constant speed	Move	Dis
G	Ground idle	Static	Dur
H	Flight idle	Static	Dur
I	Hover in ground effect	Static	Dur
J	Hover out of ground effect	Static	Dur
V	Vertical ascent in ground effect	Static	Dur Alt
W	Vertical ascent out of ground effect	Static	Dur Alt
Y	Vertical descent in ground effect	Static	Dur Alt
Z	Vertical descent out of ground effect	Static	Dur Alt
B	Approach with horizontal deceleration	Move	Dis Spd
C	Approach with descending deceleration	Move	Dis Alt Spd
E	Depart with horizontal acceleration	Move	Dis Spd
F	Depart with climbing acceleration	Move	Dis Alt Spd
T	Taxi at constant speed	Move	Spd
S	Start altitude at constant speed	--	Alt spd

Parameter values are defined as below:

PARAM	Full Name
DIST	Distance (ft)
DUR	Duration (s)
ALT	Altitude (ft AFE)
Spd	Airspeed (kts)

6.2.2.1 Additional Helicopter Step Type Information

Step Type	Description
Start Altitude	This step is used to start a profile at a given altitude and speed. Input the starting altitude and speed.
Level Fly	This step is used to maintain altitude and speed for a given distance. Input the track distance covered by the step. Altitude and speed are defined by the previous step.
App Const Speed	This step is used to descend at constant speed to a given altitude over a given distance. Input the track distance covered by the step and the final altitude. The initial altitude and speed are defined by the previous step.
App Desc Decel	This step is used to descend and decelerate to a final altitude and speed over a given distance. Input the track distance covered by the step, the final altitude, and the final speed. The initial altitude and speed are defined by the previous step.
App Horiz Decel	This step is used to decelerate to a final speed at constant altitude over a given distance. Input the track distance covered by the step and the final speed. The altitude and initial speed are defined by the previous step.
App Vertical	This step is used to maintain horizontal position while descending to a final altitude over a given duration. Input the duration of the step and the final altitude. The horizontal position of the step is calculated from the previous step and the horizontal speed is zero.
Hover	This step is used to maintain altitude and horizontal position for a given duration. Input the duration of the step. The altitude is defined by the previous step, the horizontal position of the step is calculated from the previous step, and the horizontal speed is zero.
Ground Idle	This step is used to maintain ground idle for a given duration. Input the duration of the step. The altitude is zero, the horizontal position of the step is calculated from the previous step, and the horizontal speed is zero.
Flight Idle	This step is used to maintain flight idle for a given duration. Input the duration of the step. The altitude is zero, the horizontal position of the step is calculated from the previous step, and the horizontal speed is zero.
Dep Vertical	This step is used to maintain horizontal position while ascending to a final altitude over a given duration. Input the duration of the step and the final altitude. The horizontal position of the step is calculated from the previous step and the horizontal speed is zero.
Dep Horiz Accel	This step is used to accelerate to a final speed over a given distance. Input the track distance covered by the step and the final speed. The altitude and initial speed are defined by the previous step.

Step Type	Description
Dep Climb Accel	This step is used to climb and accelerate to a final altitude and speed over a given distance. Input the track distance covered by the step, the final altitude, and the final speed. The initial altitude and speed are defined by the previous step.
Dep Const Speed	This step is used to climb at constant speed to a given altitude over a given distance. Input the track distance covered by the step and the final altitude. The initial altitude and speed are defined by the previous step.
Taxi	This step is used to taxi at a given speed. Input the speed. The track distance is calculated based on the assigned taxi ground track, and the altitude is defined by the previous step. Helicopters defined as not having wheels must taxi at an altitude greater than zero.

Helicopter procedure steps explicitly define a helicopter's flight path. There are no thrust calculations for helicopter flight paths as there are for fixed-wing aircraft. Rather, each procedure step correlates with a helicopter flight operational mode and each mode has its own set of NPD data.

Some helicopter procedure steps correlate with different helicopter flight operational modes (and therefore different NPD and directivity data) depending on their altitude. When constructing flight paths with the Hover, DepVertical, and App Vertical procedure steps, AEDT 2a calculates a ground effect altitude as follows:

Ground Effect Altitude = 1.5 x Main Rotor Diameter

If the procedure step stays below the ground effect altitude, the procedure step correlates with the corresponding In Ground Effect flight operational mode. If the step stays at or above the ground effect altitude the procedure correlates with the corresponding Out of Ground Effect flight operational mode. If a given Dep Vertical or App Vertical procedure step crosses the ground effect altitude, AEDT 2a automatically divides the step into two at the ground effect altitude and assigns flight operational modes to the two steps as appropriate.

6.2.3 How to Build a Helicopter Approach Profile

Helicopter approach profiles can be much more dynamic than fixed-wing airplane approach profiles. There are many more ways to operate a helicopter than there are to operate an airplane. AEDT 2a provides a standard approach profile for each helicopter in the database, however these standard profiles may not be appropriate for all helicopter modeling. Additionally, general guidelines are not as appropriate for helicopter approach operations as they are for fixed-wing airplanes. It is strongly recommended to evaluate the helicopter flight operations being modeled to determine if using the standard AEDT 2a helicopter procedures is appropriate. In most cases consulting with helicopter operators to design helicopter profiles that are appropriate for your study is needed. All helicopter approach profiles must start with a Start Altitude step.

For reference, AEDT 2a standard helicopter approach procedures consist of the following procedure steps:

1. Start Altitude, with altitude set to 1000 feet AFE and speed set equal to the helicopter's level reference speed.

2. Level Fly, with distance set to 87250.0 ft (approximately 14 nautical miles).

3. App Horiz Decel, maintaining an altitude of 1000 ft while decelerating to the helicopter's approach reference speed over a distance of 5000 ft.

4. App Const Speed, maintaining the helicopter's approach reference speed while descending to an altitude of 500 feet AFE over a track distance of 4800 feet.

5. App Desc Decel, descending to an altitude of 15 feet AFE while decelerating to a speed of 0 knots over a distance of 2850 feet.

6. App Vertical, maintaining horizontal position while descending to 0 feet AFE over a duration of 3 seconds.

7. Flight Idle for a duration of 30 seconds.

8. Ground Idle for a duration of 30 seconds.

6.2.4 How to Build a Helicopter Departure Profile

Helicopter approach profiles can be much more dynamic than fixed-wing airplane approach profiles. There are many more ways to operate a helicopter than there are to operate an airplane. AEDT 2a provides a standard approach profile for each helicopter in the database, however these standard profiles may not be appropriate for all helicopter modeling. Additionally, general guidelines are not as appropriate for helicopter approach operations as they are for fixed-wing airplanes. It is strongly recommended to evaluate the helicopter flight operations being modeled to determine if using the standard AEDT 2a helicopter procedures is appropriate. In most cases consulting with helicopter operators to design helicopter profiles that are appropriate for your study is needed.

For reference, AEDT 2a standard helicopter departure procedures consist of the following procedure steps:

1. Ground Idle for a duration of 30 seconds.

2. Flight Idle for a duration of 30 seconds.

3. Dep Vertical, maintaining horizontal position while ascending to an altitude of 15 ft AFE over a duration of 3 seconds.

4. Dep Horiz Accel, maintaining altitude while accelerating to a speed of 30 knots over a distance of 100 feet.

5. Dep Climb Accel, climbing to an altitude of 30 feet AFE while accelerating to the helicopter's depart reference speed over a distance of 500 feet.

6. Dep Const Speed, maintaining speed while climbing to an altitude of 1000 feet AFE over a track distance of 3500 feet.

7. Dep Horizontal Accel, maintaining altitude while accelerating to the helicopter's level reference speed over a track distance of 2800 feet.

8. Level Fly, with distance set to 93100 feet (approximately 15 nautical miles).

6.2.5 How to Build a Helicopter Overflight Profile

A typical helicopter overflight profile begins in the air at the start of an overflight track, follows the track, and ends in the air. Overflight profiles may include any of the steps defined in section 6.2.2 except for the Taxi step, and it must start with a Start Altitude step.

6.2.6 Helicopter Procedure Step Transitions

Procedure steps are combined in prescribed sequences. However, certain sequences are not allowed. For example, an approach profile cannot use an ascent step. Procedures must comply with the step transition diagrams provided here.

- The step transition diagrams use a simple convention to represent procedures:
- Ellipses represent procedure steps.
- Arrows represent a valid transition from one step to another.
- Arrows point in the direction of the allowed transition – e.g. you can go from
- Flight, Idle to Ground, Idle on an approach, but not back.
- A double sided arrow means that the transition is valid in both directions.
- An arrow looping back to a step indicates that the step can be repeated.
- A box surrounding two or more steps is used to simplify the diagram.
- Arrows connected to the box apply to each step within.
- Each step within the box can transition to any other within the box. However, speeds and altitudes must be compatible. For example, on an approach a transition from an App.Horiz.Decel step to a Hover step is valid only when the App.Horiz.Decel step has a speed of 0 knots.

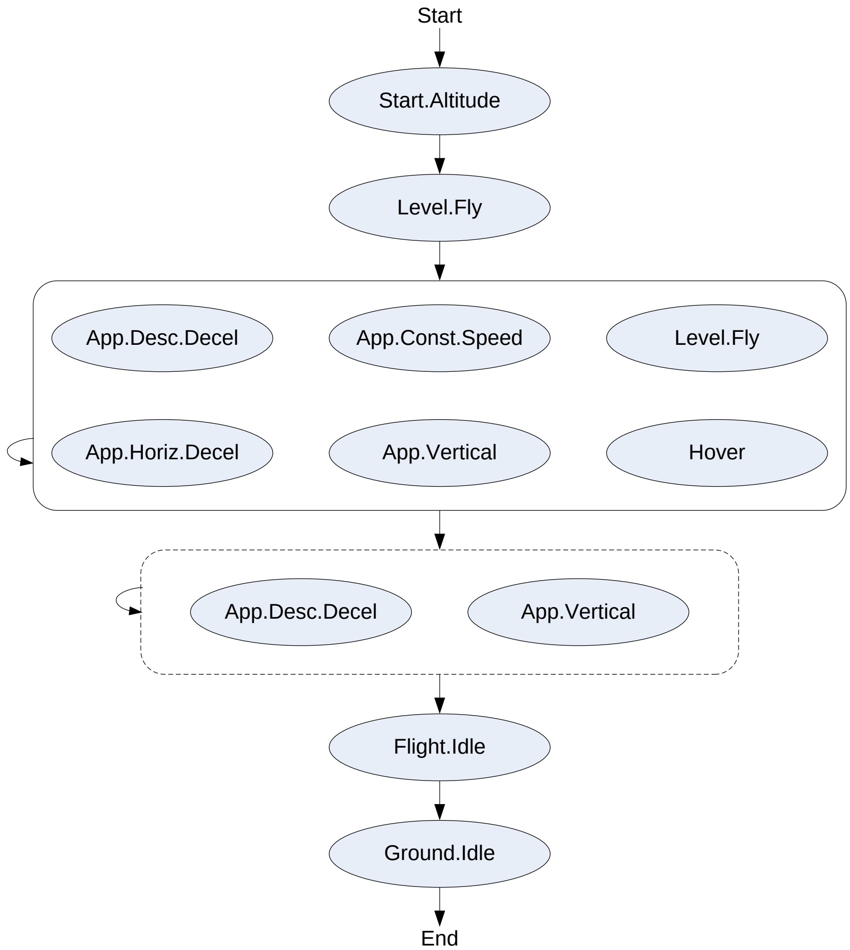


Figure 6: Helicopter Approach Step Transition Diagram

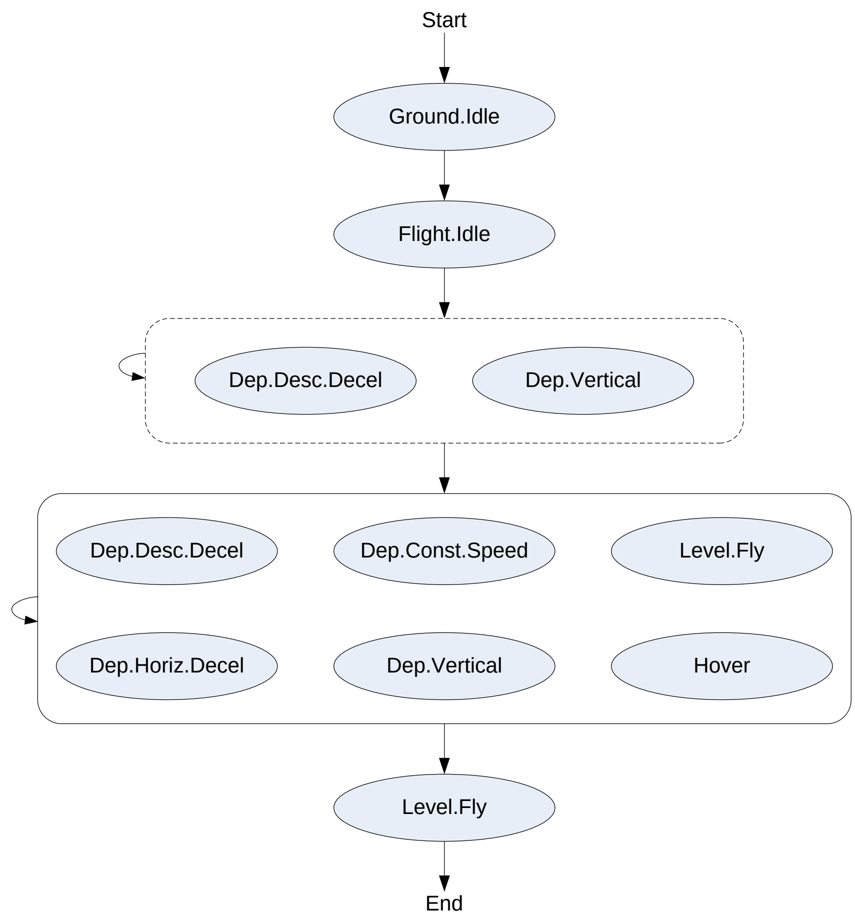


Figure 7: Helicopter Departure Step Transition Diagram

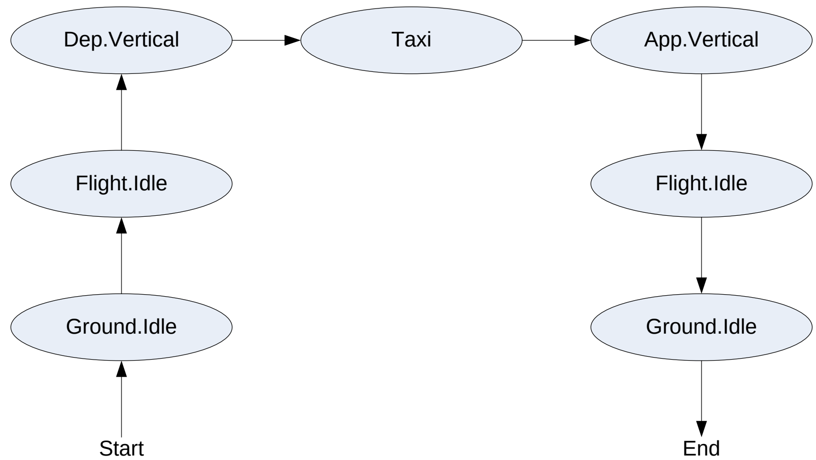


Figure 8: Helicopter Taxi Transition Diagram (not supported in AEDT 2a)

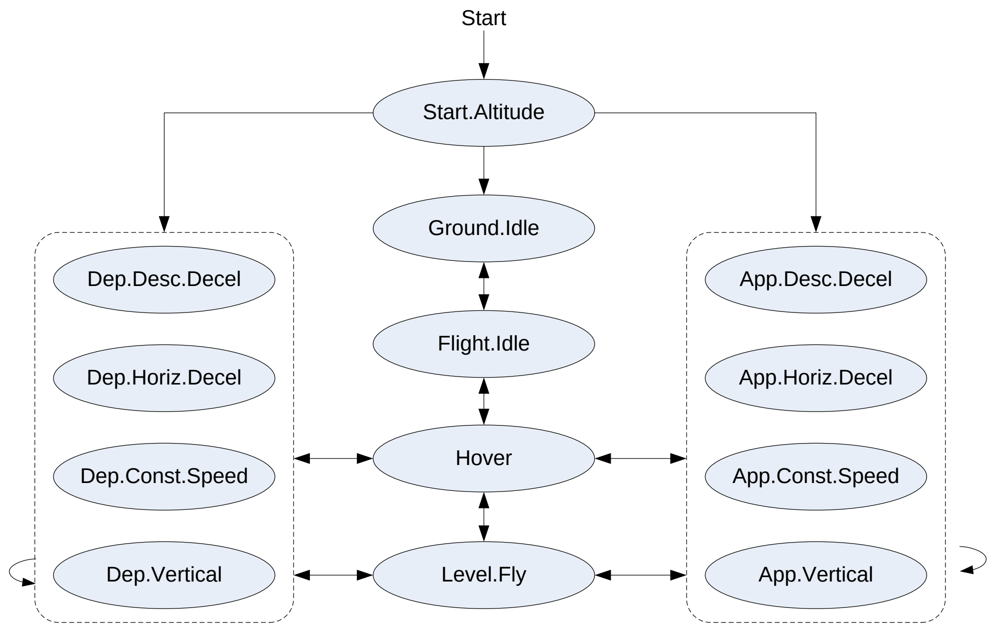


Figure 9: Helicopter Overflight Step Transition Diagram

7 XML Schema – ASIF.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="xs3p.xsl"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" att
ributeFormDefault="unqualified" version="1.1.10">
  <!-- CHANGES
    - Version 1.1.10
        - removed rotation from grid
        - removed aircraft substitutions
    - Version 1.1.9
        - added cruiseAltitude to operation
        - fixed type: opType value ToughAndGo renamed to TouchAndGo
        - updated <runway> length and width types to short from the double type
        - added airport_layout/peakMonthAverageDayScalingFactor
    - Version 1.1.8
        - added elevation and coord2DGroup to airportLayoutType
        -
 removed xyReference (and related x,y elements), trajectory, trajNode and flight elements
        - updated <runwayEnd> name to string8 from string4
        -
 added <tracks> to airport layout to allow tracks to be specified at the layout level to
avoid duplicate track generation
        -
 added <trackref> to trackOpSet which allows an operation set to reference an existing tr
ack
        - added <cases> to scenario to contain the top level <case> element
        - added airportLayouts to scenario
        - added airportLayouts and airportLayout elements to study
        - renamed icaoCode in airportLayout to airportCode
        -
 updated all airport element references to reference airportCode which allows the ability
 to specify an airport type through a "type" attribute
           (defaults to ANY airport code to support LEGACY studies)
        - removed blockPartFlag from population
        - added value to opType pattern (W - RunwayToRunway)
        - dropped arrival/departure airport info from track (not needed in 2A)
        - changed runwayEnd to runwayEndType
        - added altitude to trackSensor
        -
 added aircraftSubstitution, airport and studyBoundary content attribute enumerations
        - removed anpAirplane.thrustRestore
        - renamed aircraft.studyDescription to aircraft.description
        - renamed anpNoiseGroup.netThrustPerEngine to anpNoiseGroup.thrustSettingType
        - renamed anpAirplane.anpProfiles to profiles, anpProfiles.anpProfile to profile
        -
 renamed badaAirplane.badaProfiles to profiles, badaProfiles.badaProfile to profile
        - changed anpProcedureStep type from int to string1
        - renamed aircraft to airplane
        - renamed baseAircraft to baseAirplane
        - removed non-supported fields in anpAirplane and badaAirplane
        - moved profiles into anpAirplane and badaAirplane
        - moved anpNoiseGroup into fleet
```

```
        - deleted airframe and airframeModel from aircraft
        - moved user defined aircraft elements to before Types section
        - moved fleet element into user defined aircraft section
        - added additional string types
        - moved user defined references to types
        - refactored aircraft block to handle anp and bada airplanes in separate blocks
        - changed badaAirplane.minPayloadMass to badaAirplane.maxPayloadMass
        - added case, scenario, fleet and trackOpSet to content attribute enumerations
        - changed airoperation/flightNumber from string8 to string16
    - Version 1.1.8 changes from 1.1.7
        - moved options up one level
        - removed asif: namespace in XSD - not needed
         - Version 1.1.6 changes from 1.1.5
                  - added 'aircraft' element
        - added support for copying system runways
        - added support for helicopter flight
         - Version 1.1.5 changes from 1.1.4
                  - made runway a required element of airport.
                  -
 fixed nodeControlType attribute restriction from "integer" to "string" since it's allowi
ng string values "None/Match/etc.." in addition to the numericode control code of "0/2/et
c.."

         - Version 1.1.4 changes from 1.1.3
                  - changed the top level element from 'asif' to 'AsifXml'.
                  - removed the 'jobId' element from the annualizationCase block.

         - Version 1.1.3 changes from 1.1.2
                  - added 'content' required enumerated attribute to 'asif' element.
                  - added 'caseId' to 'case' element
                  - modified case references to point to case id.
                           - changes 'refCase' to 'refCaseId'
                           - removed 'refScenario' as it's no longer needed

         - Version 1.1.2 changes from 1.1.1
                  -
 switched order of 'startTime' and 'duration' in 'scenario' and 'case' elements.
                  - added 'blockPartFlag' to 'centroid' element.
                  - added 'climate' element and added 'climateId' sub-
element to 'case' element.
                  - added 'runway' and 'runwayEnd' sub-elements to the 'airport' element.
                  - added 'jobId' element to the 'annualizationCase' block.
                  -
 added SAE and BADA profile types: 'saeProfile', 'saeProfiles', 'badaProfile', and 'badaP
rofiles' elements.

         - Version 1.1.1 changes from 1.1
                  -
 all the string types now have minOccurs=0 for the number of characters (was 1 before).
                  -
 the 'userType' element in the 'operation' element now has 'minOccurs=0'.
                  -
 the 'reference' element within the 'case' element had it's subelement names changed
                           - 'scenario' to 'refScenario'
                           - 'case' to 'refCase'
```

```
                  -
 the 'case' element can now contain 'case' elements (i.e. cases within cases.)
                  -
 scenario element has 'startTime' type changed from 'date' to 'dateTime'.
                  - removed support for dispersed tracks
                  - changed track.name from string16 to string64
                  - annualization can be a top level block.
                  -
 added annualizationGroupCase group and made that the main sub element of annualizationGr
oup
                  - changed 'weight' attribute of annualizationGroup to an element.
                  -
 change 'weight' and 'name' attributes of annualizationCase to elements.
                  -
 removed receptor 'boundaryGrid' and 'location' elements, do these later on.
                  -
 changed 'centroid' element to have 'stateFips', 'countyFips', 'blockId', and 'bnaId' ins
tead of just a plain 'id' field.
-->
  <!--============================================================-->
  <!-- Elements                                                   -->
  <!--============================================================-->
  <xs:element name="AsifXml">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="options" minOccurs="0" maxOccurs="1"/>
        <xs:choice>
          <xs:element ref="airportLayouts" />
          <xs:element ref="annualization" maxOccurs="unbounded"/>
          <xs:element ref="boundary" maxOccurs="unbounded"/>
          <xs:element ref="case" maxOccurs="unbounded"/>
          <xs:element name="fleet" type="fleet" minOccurs="0" maxOccurs="1"/>
          <xs:element ref="operation" maxOccurs="unbounded"/>
          <xs:element ref="receptorSet" maxOccurs="unbounded"/>
          <xs:element ref="scenario" maxOccurs="unbounded"/>
          <xs:element ref="study"/>
          <xs:element ref="trackOpSet" maxOccurs="unbounded"/>
        </xs:choice>
      </xs:sequence>
      <xs:attribute name="version" type="string16" use="optional"/>
      <xs:attribute name="content" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="airportLayouts" />
            <xs:enumeration value="annualization"/>
            <xs:enumeration value="case"/>
            <xs:enumeration value="fleet"/>
            <xs:enumeration value="receptorSets"/>
            <xs:enumeration value="scenario"/>
            <xs:enumeration value="study"/>
            <xs:enumeration value="studyBoundary" />
            <xs:enumeration value="trackOpSet"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
```

```
  </xs:element>
  <xs:element name="study">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="string255"/>
        <xs:element name="type" type="studyType"/>
        <xs:element name="metricUnits" type="xs:boolean"/>
        <xs:element name="emissionsUnits" type="emissionsUnitsType"/>
        <xs:element name="description" type="string255" minOccurs="0"/>
        <xs:element ref="boundary" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="climate" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="airportLayouts" minOccurs="0" />
        <xs:element name="terrainFiles" type="string255" minOccurs="0"/>
        <xs:element ref="receptorSet" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="fleet" type="fleet" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="scenario" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="options">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="utmZoneDefault" type="xs:int" default="-1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="boundary">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="polygon" type="polygon2DType" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="climate">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="identifier" type="string8"/>
        <xs:element name="temperature" type="xs:float"/>
        <xs:element name="pressure" type="xs:float"/>
        <xs:element name="humidity" type="xs:float"/>
        <xs:element name="headWind" type="xs:float" minOccurs="0"/>
        <xs:element name="seaLevelPressure" type="xs:float" minOccurs="0"/>
        <xs:element name="dewPoint" type="xs:float" minOccurs="0"/>
        <xs:element name="windDirection" type="xs:float" minOccurs="0"/>
        <xs:element name="visibility" type="xs:float" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="airportCode">
    <xs:simpleContent>
      <xs:extension base="string4">
        <xs:attribute name="type" type="airportCodeType" use="optional" default="ANY"/>
        <xs:attribute name="country" type="string3" use="optional" default="ANY"/>
      </xs:extension>
    </xs:simpleContent>
```

```
  </xs:complexType>

  <xs:element name="tracks">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="track" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="airportLayoutType" >
    <xs:sequence>
      <xs:element name="name" type="string255" minOccurs="0" />
      <xs:element name="airportCode" type="airportCode"/>
      <xs:element name="startDate" type="xs:date" minOccurs = "0" />
      <xs:element name="elevation" type="xs:float" minOccurs="0" />
      <xs:element name="peakMonthAverageDayScalingFactor" type="xs:double" minOccurs="0"
default="1.0" />
      <xs:group ref="coord2DGroup" minOccurs="0" />
      <xs:element ref="runway" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="tracks" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>


  <xs:element name="runway">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="length" type="xs:short"/>
        <xs:element name="width" type="xs:short"/>
        <xs:element name="runwayEnd" type="runwayEnd"  minOccurs="1" maxOccurs="2" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="runwayEnd">
    <xs:sequence>
      <xs:element name="name" type="string8"/>
      <xs:group ref="coord2DGroup"/>
      <xs:element name="elevation" type="xs:float"/>
      <xs:element name="threshCrossHeight" type="xs:float" minOccurs="0"/>
      <xs:element name="threshElevation" type="xs:float" minOccurs="0"/>
      <xs:element name="glideSlope" type="xs:float" minOccurs="0"/>
      <xs:element name="intAltitude" type="xs:float" minOccurs="0"/>
      <xs:element name="depDispThresh" type="xs:float" minOccurs="0"/>
      <xs:element name="appDispThresh" type="xs:float" minOccurs="0"/>
      <xs:element name="percentWind" type="xs:float" minOccurs="0"/>
      <xs:element name="isHelipad" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="airportLayouts">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="layout" type="airportLayoutType" maxOccurs="unbounded"/>
      </xs:sequence>
```

```
    </xs:complexType>
  </xs:element>


  <xs:element name="scenario">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="string255"/>
        <xs:element name="startTime" type="xs:dateTime"/>
        <xs:element name="duration" type="xs:int"/>
        <xs:element name="taxiModel" type="taxiModelType"/>
        <xs:element name="acftPerfModel" type="aircraftPerformanceModelType"/>
        <xs:element name="bankAngle" type="xs:boolean"/>
        <xs:element name="altitudeCutoff" type="xs:float" default="18000" minOccurs="0"/>
        <xs:element name="sulfurConversionRate" type="xs:float" minOccurs="0"/>
        <xs:element name="fuelSulfurContent" type="xs:float" minOccurs="0"/>
        <xs:element name="description" type="string255" minOccurs="0"/>
        <xs:element name="airportLayouts" minOccurs="0" >
          <xs:complexType>
            <xs:sequence>
              <xs:element name="layout" type="string255" maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element ref="cases"/>
        <xs:element ref="annualization" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="case">
    <xs:complexType>
      <xs:choice>
        <xs:sequence>
          <xs:element name="caseId" type="xs:int"/>
          <xs:element name="name" type="string255"/>
          <xs:element name="source" type="emissionsSourceType" minOccurs="0"/>
          <!--  TODO: does ASIF need climate records defined for a Case?
                               <xs:element name="climate" minOccurs="0"/>
                   -->
          <xs:element name="startTime" type="xs:dateTime" minOccurs="0"/>
          <xs:element name="duration" type="xs:int" minOccurs="0"/>
          <xs:element name="climateId" type="string8" minOccurs="0"/>
          <xs:element name="hourlyWxFile" type="string255" minOccurs="0"/>
          <xs:element name="hourlyWxMD5" type="string16" minOccurs="0"/>
          <xs:element name="description" type="string255" minOccurs="0"/>
          <xs:choice>
            <xs:element ref="case" maxOccurs="unbounded"/>
            <xs:element ref="trackOpSet" maxOccurs="unbounded"/>
            <xs:element ref="operation" maxOccurs="unbounded"/>
          </xs:choice>
        </xs:sequence>
        <xs:element name="reference">
          <!--  This represents a reference to a previously defined case name. -->
          <xs:complexType>
            <xs:sequence>
              <xs:element name="refScenario" type="string255"/>
```

```
            <xs:element name="refCase" type="string255"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="cases">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="case" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="annualization">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="string255"/>
      <xs:element ref="annualizationGroup"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="annualizationGroup">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="weight" type="xs:double"/>
      <xs:element name="scaleFactor" type="xs:float" default="1" minOccurs="0"/>
      <xs:group ref="annualizationGroupCase" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="annualizationCase">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="string255"/>
      <xs:element name="weight" type="xs:double"/>
      <xs:element name="scaleFactor" type="xs:float" default="1" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="trackOpSet">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element ref="track" maxOccurs="unbounded"/>
        <xs:element ref="trackref"/>
        <xs:element ref="sensorPath" maxOccurs="unbounded"/>
      </xs:choice>
      <xs:element ref="operations" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="operation">
  <xs:complexType>
    <xs:sequence>
```

```
        <xs:element name="id" type="string16"/>
        <xs:element name="aircraftType" type="aircraftType"/>
        <xs:element name="cruiseAltitude" type="xs:double" minOccurs="0" />
        <xs:element name="numOperations" type="xs:double"/>
        <xs:element name="carrier" type="string4" minOccurs="0"/>
        <xs:element name="flightNumber" type="string16" minOccurs="0"/>
        <xs:element name="tailNumber" type="string8" minOccurs="0"/>
        <xs:element name="userType" type="string12" minOccurs="0"/>
        <xs:element name="userParam" type="string16" minOccurs="0"/>
        <xs:element name="departureAirport" type="airportCode" minOccurs="0"/>
        <xs:element name="departureRunway" type="string8" minOccurs="0"/>
        <xs:element name="departureGate" type="string4" minOccurs="0"/>
        <xs:element name="arrivalAirport" type="airportCode" minOccurs="0"/>
        <xs:element name="arrivalRunway" type="string8" minOccurs="0"/>
        <xs:element name="arrivalGate" type="string4" minOccurs="0"/>
        <xs:element name="offTime" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="onTime" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="enrouteStartTime" type="xs:dateTime" minOccurs="0"/>
        <xs:choice>
          <xs:element name="outTime" type="xs:dateTime" minOccurs="0"/>
          <xs:element name="taxiOutDuration" type="xs:int" minOccurs="0"/>
        </xs:choice>
        <xs:choice>
          <xs:element name="inTime" type="xs:dateTime" minOccurs="0"/>
          <xs:element name="taxiInDuration" type="xs:int" minOccurs="0"/>
        </xs:choice>
        <xs:choice>
          <xs:element name="saeProfile" type="profileType" minOccurs="0"/>
          <xs:element name="saeProfiles" type="profiles" minOccurs="0"/>
        </xs:choice>
        <xs:choice>
          <xs:element name="badaProfile" type="profileType" minOccurs="0"/>
          <xs:element name="badaProfiles" type="profiles" minOccurs="0"/>
        </xs:choice>
        <xs:choice>
          <xs:element name="stageLength" type="string1" minOccurs="0"/>
          <xs:sequence>
            <xs:element name="departureStageLength" type="string1" minOccurs="0"/>
            <xs:element name="arrivalStageLength" type="string1" minOccurs="0"/>
          </xs:sequence>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="operations">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="operation" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>


  <xs:element name="trackref">
    <xs:complexType>
```

```
      <xs:sequence>
        <xs:element name="layoutName" type="string255"/>
        <xs:element name="trackName" type="string64"/>
        <xs:element name="optype" type="opType"/>
        <xs:element name="runway" type="string8" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

   <xs:element name="track">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="id" type="string16"/>
        <xs:element name="name" type="string64" minOccurs="0"/>
        <xs:element name="optype" type="opType"/>
        <xs:element name="wingtype" type="wingType" minOccurs="0"/>
        <xs:element name="airport" type="airportCode" minOccurs="0"/>
        <xs:element name="runway" type="string8" minOccurs="0"/>
        <xs:element name="vectorCourseHelipad" type="xs:double" minOccurs="0"/>
        <xs:element ref="subtrack" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="subtrack">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="id" type="xs:int"/>
        <!-- NOTE: logic check to make sure all subtrack weights add up to 1 -->
        <xs:element name="dispersionWeight" type="xs:float"/>
        <xs:choice>
          <xs:element ref="trackVectors"/>
          <xs:element ref="trackNodes"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="trackVectors">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="trackVector" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="trackNodes">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="trackNode" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="sensorPath">
    <xs:complexType>
```

```
      <xs:sequence>
        <xs:element ref="sensorNode" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="sensorNode">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="lat" type="xs:double"/>
        <xs:element name="long" type="xs:double" />
        <xs:element name="altitude" type="xs:double" />
        <xs:element name="messageTime" type="xs:dateTime" />
        <xs:element name="sequenceNum" type="xs:int" />
        <xs:element name="speed" type="xs:float" minOccurs="0"/>
        <xs:element name="thrust" type="xs:float" minOccurs="0"/>
        <xs:element name="source" type="string255" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="trackVector">
    <xs:complexType>
      <xs:sequence>
        <xs:group ref="nodeIdGroup"/>
        <xs:element name="type" type="vectorTrackType"/>
        <xs:choice>
          <xs:element name="distance" type="xs:float"/>
          <xs:sequence>
            <xs:element name="angle" type="xs:float"/>
            <xs:element name="radius" type="xs:float"/>
          </xs:sequence>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="trackNode">
    <xs:complexType>
      <xs:sequence>
        <xs:group ref="nodeIdGroup"/>
        <xs:group ref="coord2DGroup"/>
        <xs:element name="altitude" minOccurs="0">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:float">
                <xs:attribute name="control" type="nodeControlType" use="optional"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
        <xs:element name="speed" minOccurs="0">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:float">
                <xs:attribute name="control" type="nodeControlType" use="optional"/>
              </xs:extension>
            </xs:simpleContent>
```

```
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="receptorSet">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="string255"/>
        <xs:group ref="receptorGroup"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="centroid">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="stateFips" type="xs:int"/>
        <xs:element name="countyFips" type="xs:int"/>
        <xs:element name="blockId" type="xs:int"/>
        <xs:element name="bnaId" type="string6"/>
        <xs:group ref="coord2DGroup"/>
        <xs:element name="elevation" type="xs:float" minOccurs="0"/>
        <xs:element name="count" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="grid">
    <xs:complexType>
      <xs:sequence>
        <xs:group ref="coord2DGroup"/>
        <xs:element name="elevation" type="xs:float" minOccurs="0"/>
        <xs:element name="width" type="xs:float"/>
        <xs:element name="height" type="xs:float"/>
        <xs:element name="numWidth" type="xs:int"/>
        <xs:element name="numHeight" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- User Defined Aircraft elements -->
  <xs:complexType  name="fleet">
    <xs:sequence>
      <xs:element name="anpNoiseGroup" type="anpNoiseGroup" minOccurs="0" maxOccurs="unbo
unded"/>
      <xs:element name="airplane" type="airplane"  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType  name="airplane">
    <xs:sequence>
      <!-- description field should go in AIR_OPERATION_AIRCRAFT.Description -->
      <xs:element name="description" type="string255"/>

      <xs:element name="baseAirplane" type="baseAirplane"/>

      <xs:element name="anpAirplaneInfo" minOccurs="0">
```

```
        <xs:complexType>
          <xs:sequence>
            <xs:choice>
              <!-- anpAirplaneId - link to another existing anpAirplane -->
              <xs:element name="anpAirplaneId" type="anpAirplaneId" />
              <!-- anpAirplane - create a new anpAirplane -->
              <xs:element name="anpAirplane" type="anpAirplane"/>
            </xs:choice>
          </xs:sequence>
        </xs:complexType>
      </xs:element>

      <xs:element name="badaAirplaneInfo" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:choice>
              <!-- badaAirplaneId - link to another existing badaAirplane -->
              <xs:element name="badaAirplaneId" type="badaAirplaneId" />
              <!-- badaAirplane - create a new badaAirplane -->
              <xs:element type="badaAirplane" name="badaAirplane" />
            </xs:choice>
          </xs:sequence>
        </xs:complexType>
      </xs:element>

      <xs:element name="airframeInfo" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:choice>
              <!-- airframeModel: link to an existing airframe -->
              <xs:element name="airframeModel" type="airframeModel" />
              <!-- airframe: create a new airframe -->
              <xs:element name="airframe" type="airframe" />
            </xs:choice>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="anpNoiseId" type="string12"/>

  <xs:simpleType name="anpAirplaneId">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="12"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="badaAirplaneId">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="10"/>
    </xs:restriction>
  </xs:simpleType>
```

```
  <xs:simpleType name="airframeModel">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="50"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="engineCode">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="50" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="engineModCode">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="50" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="baseAirplane">
    <xs:sequence>
      <xs:element name="anpAirplaneId" type="anpAirplaneId" />
      <xs:element name="badaAirplaneId" type="badaAirplaneId" />
      <xs:element name="airframeModel" type="airframeModel" />
      <xs:element name="engineCode" type="engineCode"/>
      <xs:element name="engineModCode" type="engineModCode" default="NONE" minOccurs="0"
/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="anpNoiseGroup">
    <xs:sequence>
      <xs:element name="noiseId" type="string12" />
      <xs:element name="spectralClassApproach" type="xs:short" minOccurs="0"/>
      <xs:element name="spectralClassDeparture" type="xs:short" minOccurs="0"/>
      <xs:element name="spectralClassAfterburner" type="xs:short"  minOccurs="0"/>
      <xs:element name="thrustSetType" type="string1" />
      <xs:element name="modelType" type="string1" />
      <xs:element name="npdCurves" type="anpNPDCurves" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="anpNPDCurves">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="npdCurve" type="anpNPDCurve"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="anpNPDCurve">
    <xs:sequence>
      <xs:element name="noiseType" type="string12"/>
      <xs:element name="opMode" type="string10"/>
      <xs:element name="netThrustPerEngine" type="xs:float"/>
      <xs:element name="L_200" type="xs:double" />
      <xs:element name="L_400" type="xs:double"  />
```

```
      <xs:element name="L_630" type="xs:double"  />
      <xs:element name="L_1000" type="xs:double" />
      <xs:element name="L_2000" type="xs:double" />
      <xs:element name="L_4000" type="xs:double" />
      <xs:element name="L_6300" type="xs:double" />
      <xs:element name="L_10000" type="xs:double" />
      <xs:element name="L_16000" type="xs:double" />
      <xs:element name="L_25000" type="xs:double" />
    </xs:sequence>
  </xs:complexType>


  <xs:complexType name="aircraftType">
    <xs:choice>
      <xs:element name="anpAircraftId" type="anpAirplaneId"/>
      <xs:sequence>
        <xs:element name="airframeModel" type="string50"/>
        <xs:element name="engineCode" type="string25"/>
        <xs:element name="engineModCode" type="engineModCode" default="NONE" minOccurs="0
"/>
      </xs:sequence>
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="airframe">
    <xs:sequence>
      <xs:element name="model" type="string50"/>
      <xs:element name="engineCount" type="xs:int"/>
      <xs:element name="engineLocation" type="string1"/>
      <xs:element name="designationCode" type="string1"/>
      <xs:element name="maxRange" type="xs:int" minOccurs="0"/>
      <xs:element name="introYear" type="xs:int" minOccurs="0"/>
      <xs:element name="euroGroupCode" type="string2" />
      <xs:element name="usageCode" type="string1" />
      <xs:element name="sizeCode" type="string1"/>
      <xs:element name="engineType" type="string1"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="anpAirplane">
    <xs:sequence>
      <xs:element name="anpAirplaneId" type="anpAirplaneId" />
      <xs:element name="description" type="string40"  minOccurs="0"/>
      <xs:element name="numberEngines" type="xs:int"  minOccurs="0"/>
      <xs:element name="maxGrossWeightTakeoff" type="xs:int"  minOccurs="0"/>
      <xs:element name="maxGrossWeightLand" type="xs:int"  minOccurs="0"/>
      <xs:element name="maxDsStop" type="xs:int"  minOccurs="0" />
      <xs:element name="thrustStatic" type="xs:int"  minOccurs="0"/>
      <xs:element name="noiseId" type="string12"  minOccurs="0"/>
      <xs:element name="minBurn" type="xs:double"  minOccurs="0"/>
      <xs:element name="profiles" type="anpProfiles" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="anpProfiles">
    <xs:sequence>
```

```
      <xs:element maxOccurs="unbounded" name="profile" type="anpProfile"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="anpProfile">
    <xs:sequence>
      <xs:element name="operationType" type="string1" />
      <xs:element name="profileGroupId" type="string8"/>
      <xs:element name="profileStageLength" type="string1"/>
      <xs:element name="weight" type="xs:int"/>
      <xs:choice>
        <xs:element type="anpProcedureSteps" name="procedureSteps" />
        <xs:element type="anpProfilePoints" name ="profilePoints" />
      </xs:choice>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="anpProfilePoints">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="point" type="anpProfilePoint"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="anpProfilePoint">
    <xs:sequence>
      <xs:element name="pointNum" type="xs:short" />
      <xs:element name="distance" type="xs:float" />
      <xs:element name="altitude" type="xs:float"/>
      <xs:element name="speed" type="xs:float" />
      <xs:element name="thrustSet" type="xs:float" />
      <xs:element name="opMode" type="string1" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="anpProcedureSteps">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="step" type="anpProcedureStep"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="anpProcedureStep">
    <xs:sequence>
      <xs:element name="stepNum" type="xs:int" />
      <xs:element name="flapId" type="string6"  minOccurs="0"  />
      <xs:element name="stepType" type="string1" />
      <xs:element name="thrustType" type="xs:string"  minOccurs="0" />
      <xs:element name="param1" type="xs:float"/>
      <xs:element name="param2" type="xs:float"/>
      <xs:element name="param3" type="xs:float"  minOccurs="0"  />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="badaAirplane">
    <xs:sequence>
      <xs:element name="badaAirplaneId" type="badaAirplaneId"/>
      <xs:element name="mfgDescription" type="string255" minOccurs="0"/>
      <xs:element name="referenceAircraftMass" type="xs:float" minOccurs="0"/>
      <xs:element name="minAircraftMass" type="xs:float" minOccurs="0"/>
      <xs:element name="maxAircraftMass" type="xs:float" minOccurs="0"/>
```

```
      <xs:element name="maxPayloadMass" type="xs:float" minOccurs="0"/>
      <xs:element name="weightGradient" type="xs:float" minOccurs="0"/>
      <xs:element name="maxOperatingSpeed" type="xs:float" minOccurs="0"/>
      <xs:element name="maxOperatingMachNumber" type="xs:float" minOccurs="0"/>
      <xs:element name="maxOperatingAltitude" type="xs:float" minOccurs="0"/>
      <xs:element name="maxAltitudeAtMaxTakeoffWeight" type="xs:float" minOccurs="0"/>
      <xs:element name="temperatureGradientOnMaximumAltitude" type="xs:float" minOccurs="
0"/>
      <xs:element name="wingSurfaceArea" type="xs:float" minOccurs="0"/>
      <xs:element name="buffetOnsetLiftCoeff" type="xs:float" minOccurs="0"/>
      <xs:element name="buffetingGradient" type="xs:float" minOccurs="0"/>
      <xs:element name="machDragCoeff" type="xs:float" minOccurs="0"/>
      <xs:element name="profiles" type="badaProfiles" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="badaProfiles">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="profile" type="badaProfile"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="badaProfile">
    <xs:sequence>
      <xs:element name="massRangeValue" type="string2"/>
      <xs:element name="companyCode1" type="string3"  minOccurs="0"/>
      <xs:element name="companyCode2" type="string2"  minOccurs="0"/>
      <xs:element name="companyName" type="string15"  minOccurs="0"/>
      <xs:element name="aircraftVersion" type="string12" />
      <xs:element name="engine" type="string12"   />
      <xs:element name="climbSpeedBelowTransitionAltitude" type="xs:short" />
      <xs:element name="climbSpeedAboveTransitionAltitude" type="xs:short" />
      <xs:element name="climbMachNumber" type="xs:double" />
      <xs:element name="cruiseSpeedBelowTransitionAltitude" type="xs:short" />
      <xs:element name="cruiseSpeedAboveTransitionAltitude" type="xs:short" />
      <xs:element name="cruiseMachNumber" type="xs:double" />
      <xs:element name="descentMachNumber" type="xs:double" />
      <xs:element name="descentSpeedUnderTransitionAltitude" type="xs:short" />
      <xs:element name="descentSpeedOverTransitionAltitude" type="xs:short" />
    </xs:sequence>
  </xs:complexType>

  <!--============================================================-->
  <!-- Groups                                                     -->
  <!--============================================================-->
  <xs:group name="annualizationGroupCase">
    <xs:choice>
      <xs:element ref="annualizationGroup" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="annualizationCase" minOccurs="0" maxOccurs="unbounded"/>
    </xs:choice>
  </xs:group>
  <xs:group name="receptorGroup">
    <xs:choice>
      <xs:element ref="centroid" maxOccurs="unbounded"/>
      <xs:element ref="grid"/>
    </xs:choice>
  </xs:group>
```

```
<xs:group name="nodeIdGroup">
  <xs:sequence>
    <xs:element name="id" type="string16" minOccurs="0"/>
    <xs:element name="description" type="string16" minOccurs="0"/>
  </xs:sequence>
</xs:group>
<xs:group name="coord2DGroup">
  <xs:choice>
    <xs:group ref="latlonCoordGroup"/>
    <xs:group ref="utmCoordGroup"/>
  </xs:choice>
</xs:group>
<xs:group name="latlonCoordGroup">
  <xs:sequence>
    <xs:choice>
      <xs:element name="latitude" type="latitudeDecimalType"/>
      <xs:element name="latitudeDMS" type="latitudeDMSType"/>
    </xs:choice>
    <xs:choice>
      <xs:element name="longitude" type="longitudeDecimalType"/>
      <xs:element name="longitudeDMS" type="longitudeDMSType"/>
    </xs:choice>
  </xs:sequence>
</xs:group>
<xs:group name="utmCoordGroup">
  <xs:sequence>
    <xs:element name="utmN" type="xs:double"/>
    <xs:element name="utmE" type="xs:double"/>
    <xs:element name="utmZone" type="xs:int" default="-1" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:group>
<xs:group name="aircraftTypeGroup">
  <xs:choice>
    <xs:element name="aircraftType" type="anpAirplaneId"/>
    <xs:element name="badaAircraftId" type="badaAirplaneId" />
    <xs:sequence>
      <xs:element name="aircraftAirframe" type="string50"/>
      <xs:element name="aircraftEngine" type="string25"/>
      <xs:element name="aircraftEngineMod" type="string50" />
    </xs:sequence>
  </xs:choice>
</xs:group>
<!--============================================================-->
<!-- Types                                                      -->
<!--============================================================-->
<xs:simpleType name="studyType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Emissions"/>
    <xs:enumeration value="Dispersion"/>
    <xs:enumeration value="Noise and Emissions"/>
    <xs:enumeration value="Noise and Dispersion"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="emissionsUnitsType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="MetricTonnes"/>
```

```
      <xs:enumeration value="Kilograms"/>
      <xs:enumeration value="Grams"/>
      <xs:enumeration value="ImperialTons"/>
      <xs:enumeration value="Pounds"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="polygon2DType">
    <xs:sequence>
      <xs:element name="vertex" minOccurs="3" maxOccurs="unbounded">
        <xs:complexType>
          <xs:group ref="coord2DGroup"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="polygon3DType">
    <xs:complexContent>
      <xs:extension base="polygon2DType">
        <xs:sequence>
          <xs:element name="ceiling" type="xs:float"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="taxiModelType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="UserSpecified"/>
      <xs:enumeration value="Sequencing"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="airportCodeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ICAO"/>
      <xs:enumeration value="IATA"/>
      <xs:enumeration value="FAA"/>
      <xs:enumeration value="OTHER"/>
      <xs:enumeration value="ANY"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="aircraftPerformanceModelType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ICAO"/>
      <xs:enumeration value="SAE1845"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="emissionsSourceType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Container"/>
      <xs:enumeration value="Aircraft"/>
      <xs:enumeration value="GSE Population"/>
      <xs:enumeration value="Parking Facilities"/>
      <xs:enumeration value="Roadways"/>
      <xs:enumeration value="Stationary Sources"/>
      <xs:enumeration value="Training Fires"/>
    </xs:restriction>
```

```
  </xs:simpleType>
  <xs:complexType name="latitudeDecimalType">
    <xs:simpleContent>
      <xs:extension base="xs:double">
        <xs:attribute name="positive" use="optional" default="N">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="N|n|S|s"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="latitudeDMSType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]{2}[\-|:|&quot;][0-9]{2}[\-|:|&apos;][0-9]{2}(.[0-
9]{3})?[N|n|S|s]"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="longitudeDecimalType">
    <xs:simpleContent>
      <xs:extension base="xs:double">
        <xs:attribute name="positive" use="optional" default="E">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="E|e|W|w"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="longitudeDMSType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]?[0-9]{2}[\-|:|&quot;][0-9]{2}[\-|:|&apos;][0-9]{2}(.[0-
9]{3})?[E|e|W|w]"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="profiles">
    <xs:sequence>
      <xs:element name="departureProfile" type="profileType"/>
      <xs:element name="arrivalProfile" type="profileType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="profileType">
    <xs:restriction base="string8"/>
  </xs:simpleType>
  <xs:simpleType name="opType">
    <xs:restriction base="xs:string">
      <xs:pattern value="A|Arrival|D|Departure|V|Overflight|C|Circuit|T|TouchAndGo|R|Runu
p|W|RunwayToRunway"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="wingType">
    <xs:restriction base="xs:string">
```

```
      <xs:pattern value="F|FixedWing|R|RotaryWing"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="vectorTrackType">
    <xs:restriction base="xs:string">
      <xs:pattern value="S|Straight|L|LeftTurn|R|RightTurn"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="nodeControlType">
    <xs:restriction base="xs:string">
      <xs:pattern value="0|None|1|AtOrBelow|2|Match|3|AtOrAbove"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string255">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="255"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string64">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string50">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="50"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string40">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="40"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string25">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="25"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string16">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="16"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string15">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="15"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string12">
```

```
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="12"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string10">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="10"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string8">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="8"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string6">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="6"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string4">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="4"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string3">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="3"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string2">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="2"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="string1">
    <xs:restriction base="xs:string">
      <xs:minLength value="0"/>
      <xs:maxLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

8 Splitting ASIF Files

AEDT comes with a utility to split ASIF files into smaller files. These files can be imported into AEDT using the ASIF import functions described in the User Guide. The ASIF Splitter utility takes a full study ASIF file (e.g. an ASIF file used to create a new study, with the ASIFXml content attribute type of "study") and creates a set of output files. These output files include:

- A base file containing the study data and first scenario data. This file can be used to create a study with the first scenario.
- A set of scenario files containing scenario data for the remaining scenarios.
- If the *Top Level Case* setting is selected for *Split level*, then each scenario file will have the first top level case and each additional case will be split into a separate file.
- If the *Split Receptors* setting is set to *Yes*, then the receptors from the input file will be split into a separate file.

To split an ASIF file using the ASIF Splitter utility:

1. Double-click the *FAA.AEE.AEDT.ASIF.SplitAsif.exe* located in the AEDT installation directory.
2. ASIF Splitter application will open (Figure 10).
3. Enter the directory where you want your split XML files to be written under *Output Directory*.
4. Enter the path to the ASIF file you would like to split under *ASIF File Path*.
5. Select the *Split level*:
6. Scenario: This partitions the file by scenario.
7. Top Level Case: This partitions the file by the top level case when parent\child cases are present.
8. Select whether or not to split your receptors by choosing *Yes* or *No* under *Split Receptors*.
9. Click the *Split ASIF File* button.

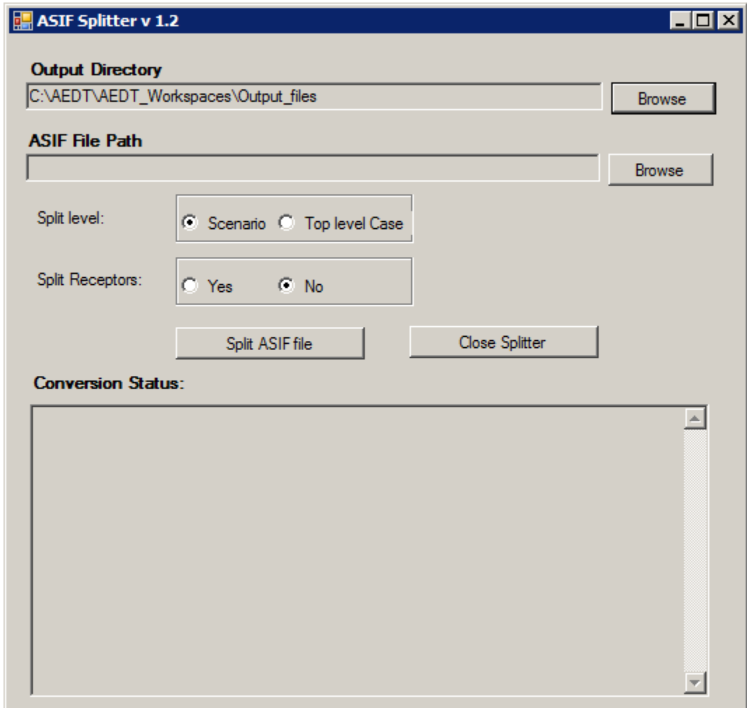


Figure 10: ASIF Splitter

10. The conversion status will be updated in the *Conversion Status* section (Figure 11). This will provide information about any errors or problems the ASIF Splitter encounters as well as alerting when the processing is complete.

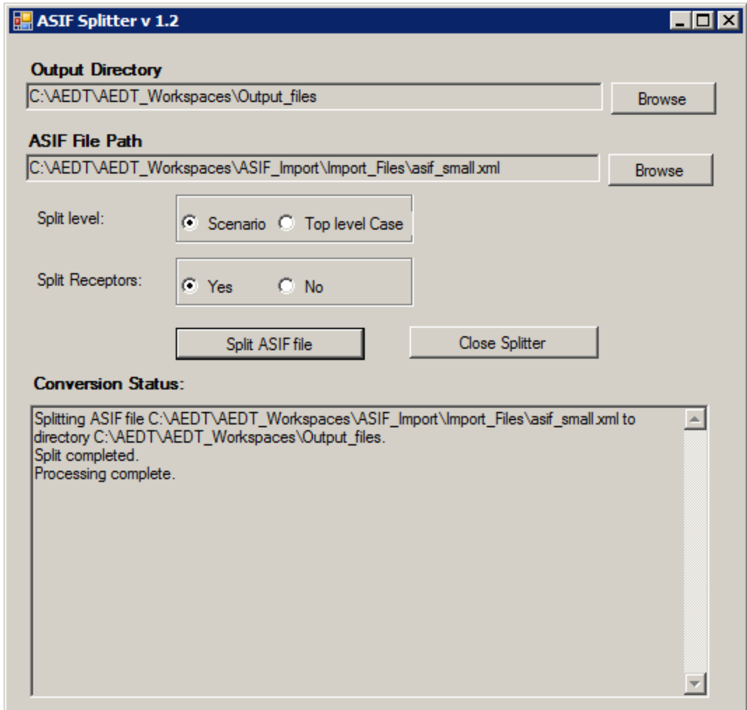


Figure 11: ASIF Splitter – Conversion Status

11. Click the *Close Splitter* button to close the ASIF Splitter application.

12. The split files will be located in the specified output directory. The file labeled "…_base_…" is the STUDY level file and must be imported first.