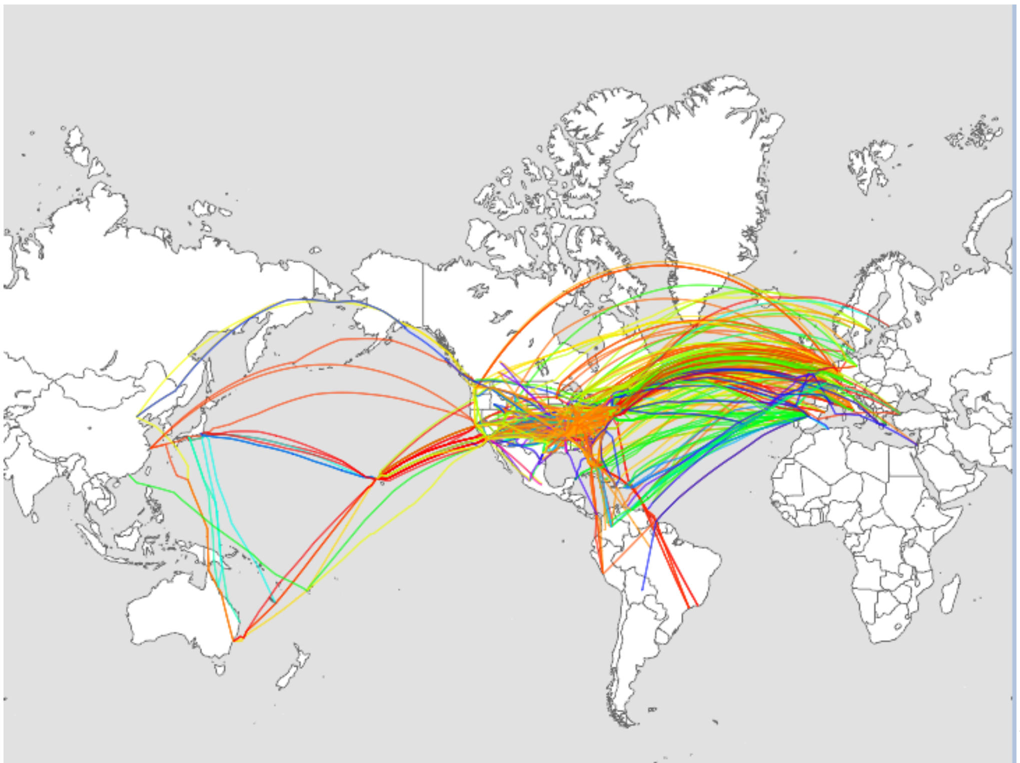
AEDT sensor path methods using BADA4



Final Report — June 2017

DOT-VNTSC-FAA-17-13

Prepared for:

Office of Environment and Energy
Federal Aviation Administration
U.S. Department of Transportation

U.S. Department of Transportation
John A. Volpe National Transportation Systems Center

Notice

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2017	Final Report

4. TITLE AND SUBTITLE	5a. FUNDING NUMBERS
AEDT sensor path methods using BADA4	FA5JA6 QC509
6. AUTHOR(S) David A. Senzig[1], Alexis Zubrow[1], Robert Downs[1], Lyle Tripp[1], Ted Thrasher[1], Joseph DiPardo[2], Mohammed Majeed[2]	5b. CONTRACT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) 1)Volpe National Transportation System Center, U.S. DOT–OST, 55 Broadway, Cambridge, Massachusetts 2) Office of Environment and Energy, FAA, 800 Independence Ave, SW, Washington, DC	8. PERFORMING ORGANIZATION REPORT NUMBER DOT-VNTSC-FAA-17-13
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) US Department of Transportation Federal Aviation Administration Office of Environment and Energy 800 Independence Ave, SW Washington, DC 20591	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION/AVAILABILITY STATEMENT This document is available on the DOT's National Transportation Library website at: http://ntlsearch.bts.gov/DOT-VNTSC-FAA-17-13	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report documents the development and use of sensor path data processing in the FAA's Aviation Environmental Design Tool (AEDT). The methods are primarily intended to assist analysts with using AEDT to determine environmental effects, such as fuel consumption and emissions, for airplane terminal area operations. The methods use the EUROCONTROL Base of Aircraft Data (BADA) family 4 as the source of the aircraft performance modeling and the fuel consumption calculations. The input data for operations are intended to come from sensor path sources, such as radar or trajectory information from simulation programs.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Aviation Environmental Design Tool, AEDT, FAA, aviation fuel consumption, aviation emissions, airplane performance	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in²	square inches	645.2	square millimeters	mm²
ft²	square feet	0.093	square meters	m²
yd²	square yard	0.836	square meters	m²
ac	acres	0.405	hectares	ha
mi²	square miles	2.59	square kilometers	km²
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft³	cubic feet	0.028	cubic meters	m³
yd³	cubic yards	0.765	cubic meters	m³
		NOTE: volumes greater than 1000 L shall be shown in m³		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
oz	ounces	28.35	grams	g
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m²	cd/m²
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in²	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm²	square millimeters	0.0016	square inches	in²
m²	square meters	10.764	square feet	ft²
m²	square meters	1.195	square yards	yd²
ha	hectares	2.47	acres	ac
km²	square kilometers	0.386	square miles	mi²
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m³	cubic meters	35.314	cubic feet	ft³
m³	cubic meters	1.307	cubic yards	yd³
mL	milliliters	0.034	fluid ounces	fl oz
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
g	grams	0.035	ounces	oz
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m²	candela/m²	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	Kilopascals	0.145	poundforce per square inch	lbf/in²

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380. (Revised March 2003)

Acknowledgments

We thank the software development team at Volpe for their work in developing the BADA4 Sensor Path Implementation: Slava Gorshkov for his work with the physics, Yasunari Tosa for his work with the MERRA-2 data, Yefim Keselman for his work with the smoothing and filtering, and Vyach Mayorskiy for program management.

We thank Sarav Arunachalam and Jared Bowden of UNC for their help and advice with incorporating detailed weather. We thank Eric Dinges and Michael Yaworski for their help and advice on the smoothing and filtering methods.

We also thank Bao Tong of the FAA for his work in developing the original smoothing and filtering methods.

Last, but not least, we thank Angela Nuic, Vincent Mouillet, and the EUROCONTROL team for developing and supporting the BADA models.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Term
ADS-B	Automatic Dependent Surveillance - Broadcast
AEDT	Aviation Environmental Design Tool
AEE	FAA's Office of Environment and Energy
AGL	Above Ground Level
AIR	Aerospace Information Report
ANP	Aircraft Noise and Performance
BADA	Base of Aircraft Data
CAEP	Committee on Aviation Environmental Protection
ETMS	Enhanced Traffic Management System
FDR	Flight Data Recorder
GNSS	Global Navigation Satellite System
ICAO	International Civil Aviation Organization
INM	Integrated Noise Model
ISA	International Standard Atmosphere
MERRA	Modern-era Retrospective analysis for Research and Applications
PBN	Performance Based Navigation
PDARS	Performance Data Analysis and Reporting System
RDP	Ramer-Douglas-Peucker filtering method
SAE	Society of Automotive Engineers
TARGETS	Terminal Area Route Generation and Traffic Simulation
TEM	Total Energy Model
UI	User Interface

Executive Summary

The Federal Aviation Administration's (FAA's) Aviation Environmental Design Tool (AEDT) allows users to determine the environmental effects of aviation. These effects include noise, fuel consumption, and emissions. The current version of the model has been shown to accurately determine noise in the terminal area, and fuel consumption for average en-route conditions, and conforms with the best practices for emission calculations. The current, public version of the model has a gap in determining fuel consumption and emissions in the terminal and en-route climbs and descents: the current version of the model uses EUROCONTROL's Base of Aircraft Data – Family 3 (BADA3) data and methods which are optimized for en-route cruise, not for the terminal area and en-route climbs and descents.

In addition to the BADA3 data and methods, the current version of AEDT also uses the data and methods of the Aircraft Noise and Performance (ANP) database. The ANP data and methods are used in the terminal area for all aspects of aircraft performance except fuel consumption. For terminal area fuel consumption, users of AEDT have the option of using the BADA3 fuel consumption methods or an internally-developed AEDT method for some aircraft. The internally-developed AEDT fuel consumption method improves the fuel consumption in the terminal area, but is not intended for use in the en-route climbs and descents.

EUROCONTROL developed the high fidelity Base of Aircraft Data – Family 4 (BADA4) data and methods to address some of the shortcomings of the BADA3 data and methods. In 2014, the FAA and Volpe incorporated the ANP and the BADA4 models into a version of AEDT used for analyses conducted in support of studies by the International Civil Aviation Organization (ICAO). That version of AEDT used the ANP standard procedural data in the terminal areas and used BADA4 in the en-route areas of the modeled flights. Successful use of that version of AEDT in the ICAO use case prompted the FAA's Office of Environment and Energy (AEE) and Volpe staff to consider using the BADA4 data and methods for the more general sensor path case – instead of modeling fixed standard procedures, the detailed flight trajectory (the sensor path) of any particular operation could be used to solve the equations of motion, and hence determine the thrust and fuel consumption along the flight trajectory. The version of AEDT which contains the BADA4 data and methods is not yet available to the general public. AEE plans to include BADA4 in a future public release of AEDT.

This document discusses using BADA4 data and method in AEDT with sensor path trajectory data. AEE proposed the use of these new data and methods explicitly to assist the Office of the Next Generation Air Traffic Control System (NextGen) in analyses of Performance Based Navigation (PBN) procedures in the terminal area. In particular, staff from the NextGen office have stated that fuel consumption is the primary metric used to adjudicate the value of PBN procedures – this new version of AEDT is explicitly focused on assisting with those types of analyses.

The sensor path data discussed in this report are aircraft position and time data that could come from existing radar sources, simulation data, or any future source which contains aircraft position information (e.g., Automatic Dependent Surveillance data). Use of 'noisy' sensor path data in AEDT's BADA4 Sensor Path Implementation requires pre-processing of the data to ensure that small anomalies in the data do not distort the fuel consumption results. The pre-processing steps involve smoothing and filtering the input data; these steps are discussed in the body of the report.

Testing has shown that local atmospheric effects (e.g. winds and temperature) influence the accuracy of modeling individual flights. As part of the BADA4 Sensor Path Implementation work, AEDT's high-fidelity weather model was extended to include support for a state-of-the-art data source, the Modern-Era Retrospective analysis for Research and Applications, Version 2 (MERRA-2). For the data set used in this analysis, using the MERRA-2 data significantly reduced the difference between the modeled and the aircraft-reported fuel consumption; this difference reduction is discussed in the body of this report.

This report includes a validation and verification (V&V) section where we compare the results of the BADA4 Sensor Path Implementation against different sets of Flight Data Recorder (FDR) information. This comparison showed that one of the most important parameters required for accurate fuel consumption prediction is accurate aircraft weight. While accurate weights are required for accurate total fuel consumption, inaccurate weight estimates may not be crucial if differences in alternatives are compared. The same is true with atmospheric effects. When weights and atmospheric information are known, as they are for the V&V data set, the differences between the FDR and the AEDT BADA44 Sensor Path Implementation fuel consumption are on the order of 1%.

Finally, we note that the philosophy of the development of the program has been to allow the user flexibility in the input data. To the extent possible, AEDT's BADA4 Sensor Path Implementation will process the sensor path data provided by the user – as with any computer model, unrealistic inputs will lead to unrealistic outputs.

1. Introduction

The Aviation Environmental Design Tool (AEDT) is the Federal Aviation Administration's (FAA's) state-of-the-art software tool for evaluating the environmental effects of aviation operations (Federal Aviation Administration, 2016). As of this writing, the current AEDT version (AEDT 2C SP2) uses EUROCONTROL's Base of Aircraft Data family 3 (BADA3) methods and data (EUROCONTROL Experimental Centre, 2014) to determine aircraft performance, including fuel consumption, for aircraft operations in the en-route regions of flight. BADA3 works well in the en-route cruise region of flight, but is not optimized for terminal area operations, off-design cruise, or en-route climbs and descents.

In addition to the use of BADA3 data and methods for en-route regions, the current version of AEDT also uses the data and methods of the Aircraft Noise and Performance (ANP) database. The ANP data and methods are used in the terminal area for all aspects of aircraft performance except fuel consumption. For terminal area fuel consumption, users of AEDT have the option of using the BADA3 fuel consumption methods or an internally-developed AEDT fuel consumption method (Senzig, Fleming, & Iovinelli, 2009)(SFI) for some aircraft. The SFI fuel consumption method improves the fuel consumption in the terminal area, but is not intended for use in the en-route regions.

EUROCONTROL developed the high fidelity Base of Aircraft Data – Family 4 (BADA4) data and methods to address some of the shortcomings of the BADA3 data and methods. BADA4 contains high fidelity aircraft performance data for all regions of flight. In 2014, the FAA and Volpe incorporated the ANP and the BADA4 models into a version of AEDT used for analyses conducted in support of studies by the Committee on Aviation Environmental Protection (CAEP) of the International Civil Aviation Organization (ICAO). That CAEP BADA4 Implementation version of AEDT used the ANP standard procedural data in the terminal areas and used BADA4 in the en-route regions of the modeled flights.

The ANP database represents the legacy Integrated Noise Model (INM) and AEDT terminal area methods and data; these methods and data were originally intended for use in airport noise modeling. The CAEP BADA4 Implementation can use either the existing AEDT terminal methods or the BADA4 methods for fuel consumption in the terminal area. The terminal performance methods in the CAEP BADA4 Implementation use the ANP procedure step methods while the en-route climb, cruise, and en-route descent regions use the BADA4 standard methods. The CAEP BADA4 Implementation follows the given flight's trajectory path (i.e. the reported points of latitude and longitude of the flight), but not the altitude or airspeed of the trajectory (except for the maximum altitude in the trajectory, which defines the upper limit of the modeled cruise altitude).

To make BADA4 in AEDT more widely useful than the procedure-based, inventory-centric methods of the CAEP BADA4 Implementation, and to specifically support analyses by the Office of the Next Generation Air Traffic Control System (NextGen), AEE tasked Volpe to expand the BADA4 implementation to include the 'sensor path' methods of the standard AEDT with BADA3. Sensor path methods use the physical change of state and position of the modeled aircraft, as reported by a particular sensor system, to determine the aircraft's required thrust and, consequently, fuel consumption and emissions. Sensor path methods are also available using BADA3 in the current version of AEDT[1]. In addition, AEE required that the new BADA4 Sensor Path Implementation make use of the new methods of detailed weather expected to be available to the user. Staff from the FAA's NextGen office have stated that fuel consumption is the primary metric used to determine the potential benefits of PBN procedures in the terminal area. The NextGen office's need for accurate fuel consumption data and methods in the terminal area for modeling new PBN procedures drove the development of this new version of AEDT.

Note that the BADA4 methods and data are proprietary to EUROCONTROL, and so are not discussed in detail in this report.

[1] The current BADA3 methods also allow the use of altitude control methods (Federal Aviation Administration, 2016).

2. BADA4 development in AEDT

Current methods

The CAEP BADA4 Implementation is the basis for the BADA4 Sensor Path Implementation. In the CAEP BADA4 Implementation, the thrust in climb conditions is known from the BADA4 methods; in cruise the thrust is set equal to the aircraft drag, and in descent the thrust is calculated based on the known glide-path angle and the characteristics of the aircraft. These methods work well for analyzing large numbers of operations, where the details of individual flights are not significant[2]. However, for studies where the details of individual flights are important, the CAEP BADA4 Implementation is not adequate.

The CAEP BADA4 Implementation relies on BADA Version 4.2, released by EUROCONTROL in April of 2016. Version 4.2 added new aircraft types and also added high-thrust data for some aircraft. While the high-thrust data improve the performance modeling in the departure region, the added data are not sufficient to model the ground roll of the aircraft[3] -- so the CAEP BADA4 Implementation uses the associated ANP ground roll data for all aircraft. The ground roll data for landing are also not available in BADA4; so again, the ANP data are used. The switch from ANP modeling to BADA4 modeling occurs during departures at the transition from ground roll to the first usable in-flight data point on departure. Conversely, on arrivals the transition occurs from the last usable in-flight data point to a point 50 feet over the threshold of the arrival runway; from the threshold crossing point, the ANP data are used for modeling the landing and roll-out.

BADA4 Sensor Path Implementation

In contrast to the CAEP analyst, sensor path users are generally interested in the details of particular operations. For example, users of the model may be determining efficiencies of particular types of operations with a sub-set of aircraft at a single airport. The data requirements for sensor path modeling will be much greater than for inventory modeling – sensor paths, by definition, require knowledge of the position and characteristics of the aircraft (its 'state') along the entire (or partial) trajectory of the operation. In the BADA4 Sensor Path Implementation, not just the current state, but the *change* in the aircraft state from one position to the next is known; the thrust required to make that state change is calculated from an energy balance.

The BADA families (3 and 4) both use a total energy method (TEM) to model aircraft performance. The

[2] In the CAEP analyses, users are interested in general trends as they potentially need to compare large amounts of data, perhaps across several analysis periods.
[3] The BADA data and methods only model in-flight conditions – they can't be used for ground operations, such as taxi and ground roll.

BADA TEM uses a difference in energy added to (or subtracted from) the aircraft via the difference in thrust and drag to determine the change in kinetic and/or potential energy of the aircraft. In the BADA4 Sensor Path Implementation, the reverse situation is used: the change in kinetic and/or potential energy from one sensor point to the next is used, together with the calculated drag, to find a thrust which will produce that change. That is, the change in the aircraft state is known from the sensor path information, and the thrust required to make that state change is calculated from the TEM.
There are four possible state changes:

1. Increasing kinetic energy, increasing potential energy (the aircraft is accelerating and climbing),
2. Increasing kinetic energy, decreasing potential energy (the aircraft is accelerating and descending),
3. decreasing kinetic energy, increasing potential energy (the aircraft is decelerating and climbing), and
4. decreasing kinetic energy, decreasing potential energy (the aircraft is decelerating and descending)

The BADA4 Sensor Path Implementation is set up so that cases where no change in state occurs between two sensor path points (such as in cruise) are handled by one of the above conditions as a limiting case.

The sensor path methods have been implemented and tested using Flight Data Recorder (FDR) information as inputs. These data were used as the primary test case since they are the best data available; we know that the aircraft was capable of flying the sensor path implicit in the data and we can use the FDR data to determine intermediate state parameters to validate the BADA4 methods. The FDR information available to the development team is limited to a sub-set of aircraft actually used in the fleet, but this is not a limitation for testing purposes.

The sensor path methods used in the current analyses use a method of trajectory following from the start of the takeoff roll to the end of the landing roll-out. This means that the BADA4 process assumes the trajectory information contains the departure and arrival runway position data.

Expected use of the BADA4 Sensor Path Implementation in AEDT

We expect that the primary use of the BADA4 Sensor Path Implementation will be:

1. processing data from ground-based systems, e.g. radar-based, such as PDARS data,
2. processing data from aircraft-based systems, e.g. either Global Navigation Satellite System (GNSS) systems like the Automatic Dependent Surveillance Broadcast (ADS-B) system or integral systems such as FDR systems, and
3. processing data from simulation systems, e.g. the Terminal Area Route Generation and Traffic Simulation (TARGETS).

Since these systems may have different flight regions of interest, AEE has asked for BADA4 to be available in the following regions:

1. starting at 10,000 feet altitude from the terminal airports,

2. starting below 10,000 feet altitude from the terminal airports, and
3. Partial paths from/to the terminal airports

Runway to runway methods

Runway to runway methods are those which start with the aircraft stopped at the beginning of the departure runway, and end with the aircraft rolling out on landing on the arrival runway. The departure ground roll is modeled using the methods of ANP; BADA4 provides no guidance for modeling aircraft on ground. The transition from the ground roll to the first sensor path point which is 500 feet above the runway is done with the BADA4 methods.

All segments from the ground-to-flight transition to the point 50 feet above the landing threshold are calculated with the methods discussed above. The last sensor path calculation on arrival is the segment from the last point above 500 feet Above Ground Level (AGL) to a point which is 50 feet above the landing threshold. From 50 feet above the landing threshold, the landing is calculated using the methods of ANP.

Partial path methods

Partial path methods are those which either stop at a particular altitude when departing or begin at a particular altitude when arriving. These partial path methods are a subset of the existing runway to runway method and are used primarily in the terminal area of the airport to determine local impacts of aviation operations without the need for the full flight trajectory.

From a processing perspective, departure partial paths merely involve stopping the flight at some pre-determined point. The departure partial path must still be provided with a starting weight which is typically based on the distance the aircraft would travel in the whole flight; the distance itself associated with the weight is also an input. Since departures start at a known elevation (i.e. the runway's) at zero speed, no additional information is required.

Arrivals require knowledge of the initial conditions of the aircraft state. In addition to the initial weight these partial paths also require the initial speed, altitude, and acceleration of the aircraft. The distance along the trajectory to the airport/runway is also a required input. The arrival process can step forward along the given trajectory to the runway once those data are known.

Smoothing and filtering

Existing methods

In the BADA4 Sensor Path Implementation, there are two legacy internal AEDT methods – both of which have historically been used in the BADA3-based sensor path workflow – that are also available to do trajectory pre-processing. One method is for smoothing and the other method is for filtering of sensor path trajectories. The existing methods are discussed in the AEDT Technical Manual (Federal Aviation Administration, 2016) in section 3.7.2.1.

New methods

The Sensor Path Import (SPI) Tool is a preprocessor geared toward preparation of four-dimensional (4-D) sensor path trajectory data for AEDT performance modeling; it can also be used to populate operational metadata in an AEDT study. The SPI Tool currently exists as a command line tool; AEE and Volpe are considering changing this to a graphic UI in the next version of AEDT.

Most operational metadata such as off-times, on-times, runway ends are typically provided by the user to the SPI Tool. The SPI Tool does have the ability to provide the user with a default aircraft engine selection for a given ICAO aircraft type[4]. For an exhaustive list of operational metadata fields, see *Appendix C: Sensor Path Import Tool*.

The SPI Tool contains both smoothing functionality and filtering functionality, the foundation of which was a prototype method developed internally by AEE using the R™ Statistical Language[5]. In the smoothing process, data noise and spurious points are removed from ground track, altitude, and ground speed data using the methods described below. Following trajectory smoothing, these data are filtered by reducing the number of points (i.e. subsampled) while retaining characteristics and so-called change points of the original trajectory. The user has the option of smoothing input ground speeds or recalculating the speeds based on smoothed longitude/latitude coordinates. Additionally, filtering for point reduction is an optional process. Finally, the sensor path import tool can be run in pass-through mode where no smoothing or filtering is applied. Thus, five trajectory data workflows exist in the smoothing and point-reductive filtering pre-processor:

1. no smoothing or point-reductive filtering methods are applied
2. smoothing using original speeds without point-reductive filtering
3. smoothing using recalculated speeds without point-reductive filtering
4. smoothing and point-reductive filtering using original speeds
5. smoothing and point-reductive filtering using recalculated speeds

The first step in the process is to smooth the trajectory in the ground plane. This smoothing is done via a cubic spline function. The data are re-sampled along the spline to retain the original ground speed data. Altitude smoothing consists of applying a Hampel filter (Pearson, Neuvo, Astola, & Gabbouj, 2016) to replace spurious points. This Hampel filter uses a point-by-point moving (i.e. overlapping) window in which values deviating from the window median by a specified amount are deemed outliers and replaced with the local median value. The width of the window and outlier criterion are user-selectable

[4] Note that not all ICAO aircraft types are covered and that more information is available in *Section 4* in the subsection entitled *Importance of Fleet mappings*.
[5] R™is a language and environment for statistical computing and graphics: https://www.r-project.org/about.html

parameters.

Similarly, a Hampel filter is applied to ground speed data in advance of the next step in the smoothing process: ground-speed smoothing. In this step, the ground speed is averaged in a *non-overlapping* moving window comprising a user-selectable number of points. As with the altitude smoothing, points rejected by the Hampel ground speed filter are replaced by the median value of the points within the current window. The user has the option to recalculate the ground speed. In the recalculate-speed workflow, ground speed is derived (rather than purely averaged) from the input times and from the splined ground track. More specifically, in the recalculate-speed workflow, the smoothed ground track data are used to compute speeds using centered finite divided differences. The arc length traversed in each window is calculated at the segment level assuming a WGS84 ellipsoid; the average window speed is then computed using the difference in time across the window endpoints and the sum of the arc-length segments. Whether or not to recalculate the ground speed data may be a function of the input data quality. For the FDR data discussed in more detail below, the data quality are high and the recalculated workflow provided no benefit. Down-sampling is also accomplished in this step, in that the average speed in each window is used to represent the speed over the corresponding time intervals, with the center point in each window making up the down-sampled times. If the filtering option is not invoked, smoothed ground speed data are linearly interpolated back onto the input times such that the number of points is retained. A flowchart of the smoothing process is shown in Figure 1; the flowchart illustrates data inputs and outputs for the functions that make up the trajectory smoothing process. The center column describes the methods applied in each function.

In the optional filtering process, the number of trajectory points is reduced using the following methods. Ground-track and altitude filtering utilize the Ramer-Douglas-Peucker (Ramer, 1972; Douglas & Peucker, 1973) method. Given an input dataset, this method seeks a new curve which minimizes both the perpendicular distance between input and output curves, and the number of points needed to represent the output curve. In the altitude filtering step, a one-dimensional Ramer-Douglas-Peucker (RDP) algorithm is applied with a user-selectable tolerance (default value: 200 feet) to down-sample the altitude data. Similarly, a two-dimensional RDP algorithm is used for ground track smoothing. The nature of these algorithms is such that points are removed if they are not needed to represent the shape of the input data. During cruise, for example, few points exist in the down-sampled altitude profile as altitude is not changing significantly. In the speed-smoothing step, sub-sampling occurs during the averaging process and those values then become the filtered speeds. As a result of applying different filtering methods, down-sampled ground tracks, altitude, and speed profiles have distinct sets of associated times. The last step in the filtering process is to generate a set of longitude / latitude / altitude / speed values at a common set of times. The filtered times are determined as the union of down-sampled times associated with each of these components of the 4-D trajectory. Linear interpolation is used to recreate values at the "missing" points in time.

Figure 2 illustrates the optional point-reductive trajectory filtering process. As in Figure 1, function inputs and outputs are given in the left and right columns, respectively. The center column describes the methods applied by each function.

The smoothing and filtering processes can be controlled by the user through a set of parameters. These parameters, their default values, and notes on their usage are provided in Table 1. An example of the inputs and outputs of the smoothing and filtering process is plotted in Figure 3. In this case, the input trajectory has 1,110 points and the filtered output has 330 points.

Table 1. User-configurable options for controlling smoothing and filtering processes

Variable name	Description	Default value	Warning	Error	Notes	Additional Treatment by R script
altWindow	Altitude smoothing: Hampel filter window width	11	val. < 1	val. < 0	Value is half of the total width: e.g. 11 points corresponds to 11+11+1=23 total points	• If non-integer, value is rounded up • If window is greater than number of points, window width set to number of points
altOutlier	Altitude smoothing: outlier criterion	3	val. < 1	val. < 0	Units are in terms of σ	
altTol	Altitude smoothing: RDP tolerance	200		val. < 0	Tolerance in ft.	
spdWindow	Speed smoothing: Hampel filter window width	11	val. < 1	val. < 0	Value is half of the total width: e.g. 11 points corresponds to 11+11+1=23 total points	• If non-integer, value is rounded up • If window width is greater than number of points, window width set to number of points
spdOutlier	Speed smoothing: outlier criterion	3	val. < 1	val. < 0	Units are in terms of σ	
spdAveWindow	Speed smoothing: averaging window width / point reduction ratio	5		val. < 1	Ratio of N:1 retains every Nth point	• If non-integer, value is rounded up • Maximum window width is limited such that at least three windows are present
distTol	Ground-track filtering: RDP tolerance	0.10		val. < 0	Tolerance in nmi	
filt_flag	Boolean switch for down-sampling	TRUE			Activates down-sampling	

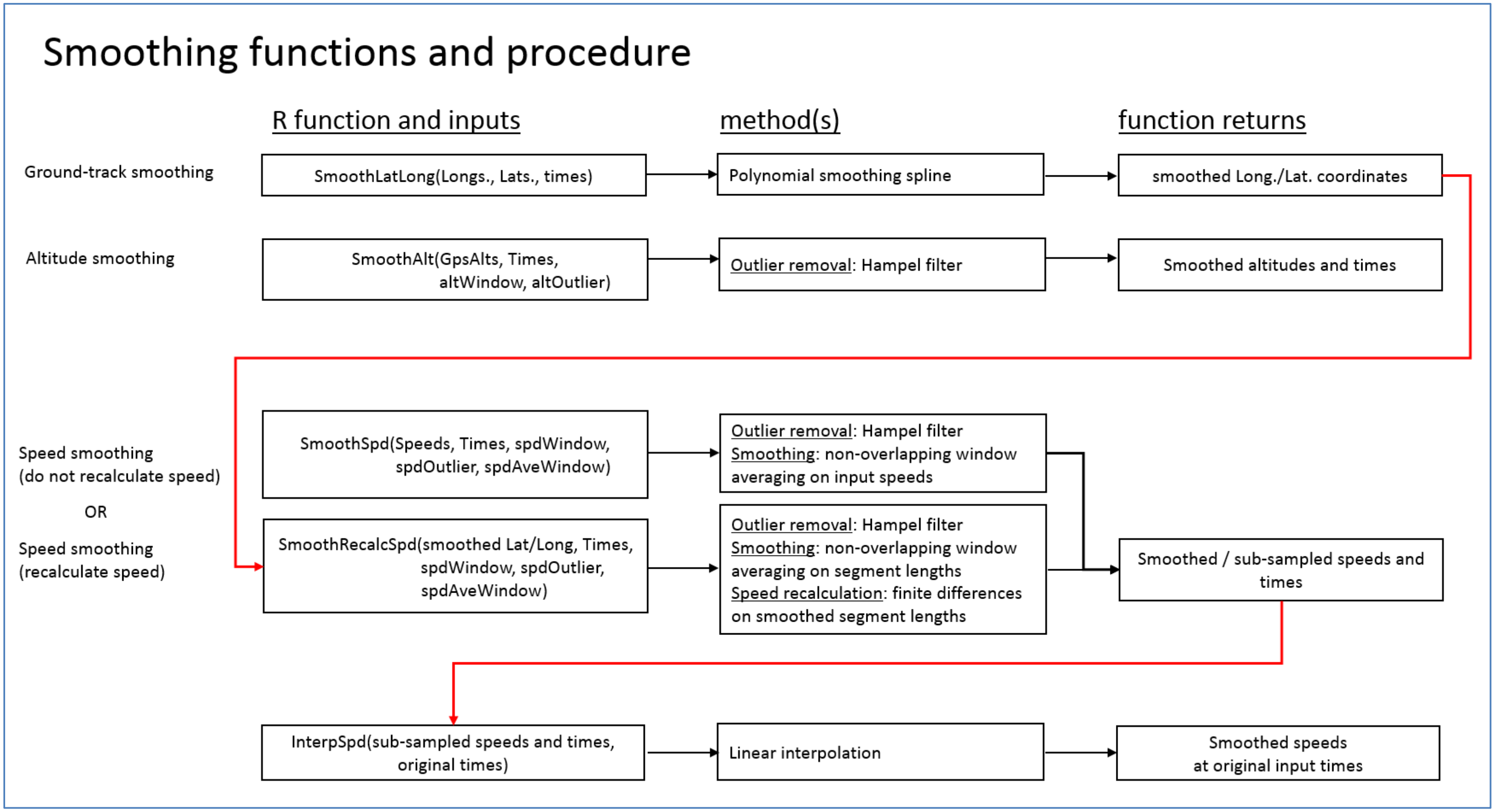


Figure 1. Smoothing process flowchart

Filtering functions and procedure

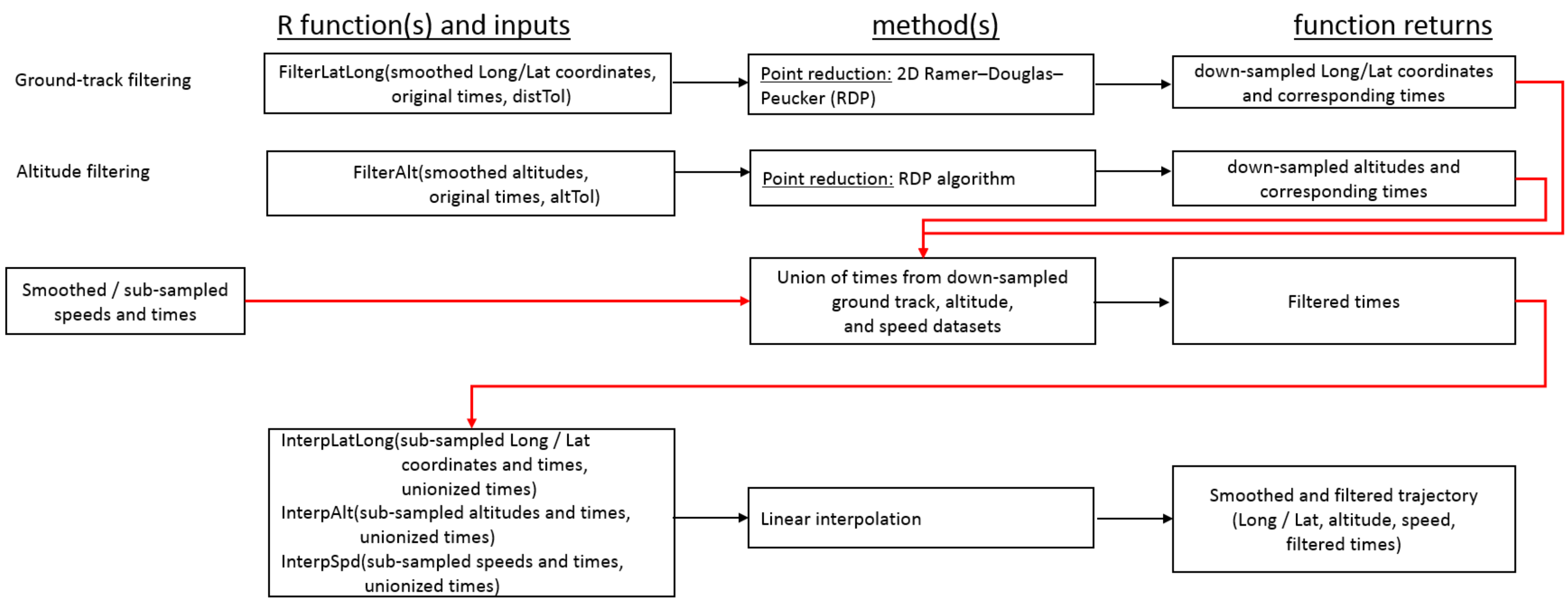


Figure 2. Filtering process flowchart

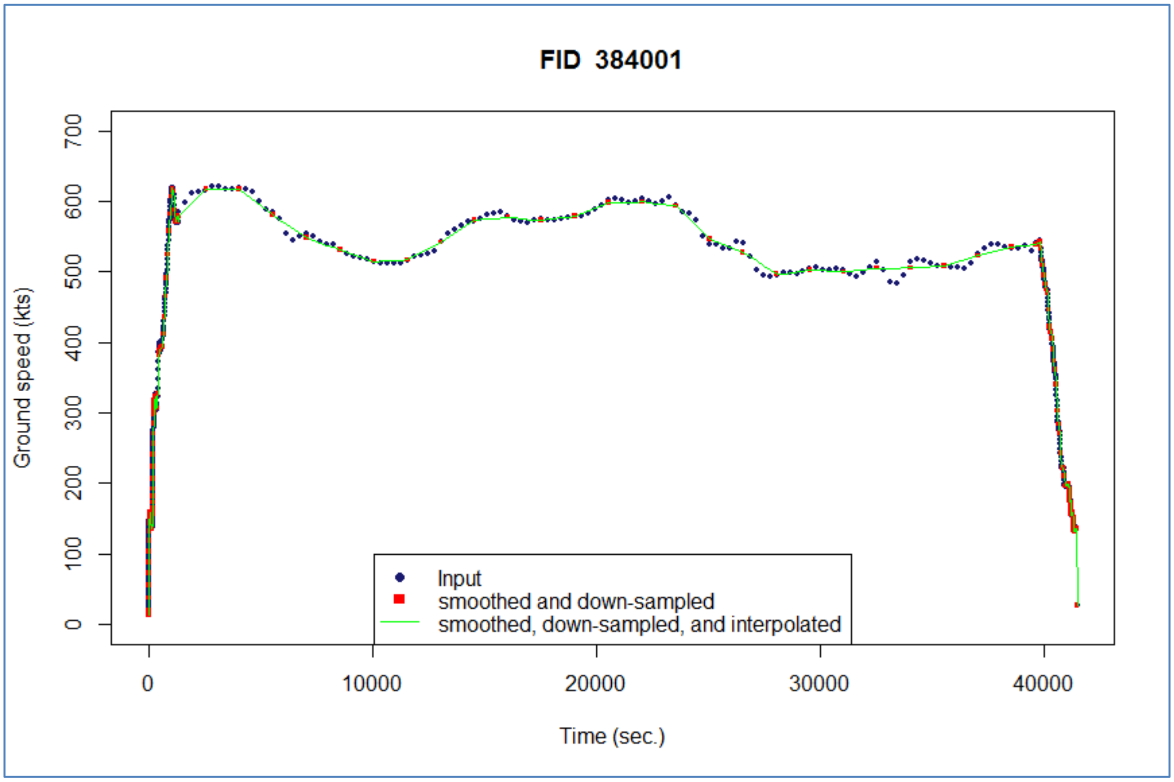


Figure 3. Example of smoothing and filtering process

High Fidelity Weather

The BADA4 implementation requires weather information (wind speed, wind direction, and atmospheric parameters of pressure, temperature, and density) at each position along the trajectory. These weather data can be populated with user-defined, International Standard Atmosphere (ISA), or high fidelity weather data. The BADA4 processes call the weather module in AEDT by reporting the current elapsed time from the start of the operation, and the position of the aircraft. This call to the AEDT weather module is made for every point in the trajectory and no weather data are retained from one point to the next.

BADA4 uses a different philosophy than the existing AEDT performance methods to track the aircraft location. The existing AEDT methods were developed from the use of the Society of Automotive Engineer's (SAE's) Aerospace Information Report (AIR) 1845 (A-21 Committee, 1986), which was the original standard for calculating aircraft noise in the vicinity of airports. SAE-AIR-1845 used a ground-centric view of the motion of the aircraft because it was focused on the noise effects of aircraft operations on receptors on the ground and the motion of the aircraft relative to the ground-based noise receptors was the important consideration. In BADA4 the motion of the aircraft relative to the air mass in which the aircraft is traveling is the important consideration; this is not done to diminish the effects of noise on the ground, but rather to start with the actual performance methods used in the real world, and then to convert the results of those performance methods back to the ground where required. This approach provides an improvement in the modeling of the aircraft performance (i.e. better fuel consumption and emission modeling) without degrading the quality of the noise[6] modeling output. Note the performance of the aircraft is air mass dependent, while the navigation of the aircraft is ground dependent; aircraft navigation, even GNSS-based, uses waypoints defined by fixed latitudes and longitudes.

BADA4 calculates the effect of the wind on the aircraft's ground track using the classic wind triangle, which is based on the law of cosines. Using vector math allows the effects of cross winds to be accounted, rather than just the headwind effects currently accounted for in the existing AEDT performance model.

The BADA4 Sensor Path Implementation effort focused on the usage of the Modern-Era Retrospective analysis for Research and Applications, Version 2 (MERRA-2) dataset as the high fidelity weather source. Support for MERRA-2 was developed because it is a state-of-the-science atmospheric reanalysis product, at the time of this document, which has the following characteristics: global coverage, high resolution in the horizontal dimension, high resolution in the vertical dimension, and the results covered a long time period (1980 – present). Other detailed weather sources can also be used with the BADA4 Sensor Path Implementation. These other weather sources are discussed in section 2.2.3 of the AEDT Technical Manual. Appendix D discusses methods of importing the MERRA-2 weather data. Note that the user can import both 3D weather (the data above the surface) and the 2D weather data (the data at the surface). We recommend importing both data sets to eliminate possible extrapolation problems at low altitudes.

User interface improvements

The CAEP BADA4 Implementation had a user interface (UI) with ambiguous settings. To reduce ambiguity, the UI was modified to group input parameters in a more logical relationship. Ambiguity in the UI was further reduced by replacing check boxes for similar choices with radio buttons, so only a single choice from a set of options is now enforced.

A screen shot below displays the following typical preferences required to exercise the BADA4 Sensor Path Implementation:

- The Workflow should be set to the "Sensor Path". The alternative workflow, "ANP Procedures and Data Below 10K feet", is also known as the CAEP workflow and described in more details in the Introduction section of this report.
- In case of the Sensor Path workflow, the "ANP Segment Fuel Model" setting affects only the

[6] Currently, noise modeling in the BADA4 Sensor Path Implemenation is still pending further validation.

runway segments of the flight. The recommended fuel setting is the “BADA4 model” which allows the user to utilize the BADA4 fuel model along the full flight and including the runway segments. The alternative setting, “SFI model”, refers to the Senzig, Fleming, Iovinelli fuel model, which is the default fuel model used by the AEDT’s APM (Aircraft Performance Module) workflow.

- The “Speed type” preference settings takes effect only if the input speed type was not provided during the import of the sensor path data into a study. The legacy import tools used at Volpe did not allow the user to set the input speed type. The Sensor Path Import Tool included with the “AEDT 2c SP2 B4” installer allows the user to set input speed type. The AEDT’s APM workflow is not affected by this preference; the APM assumes the “Ground speed” input speed type.

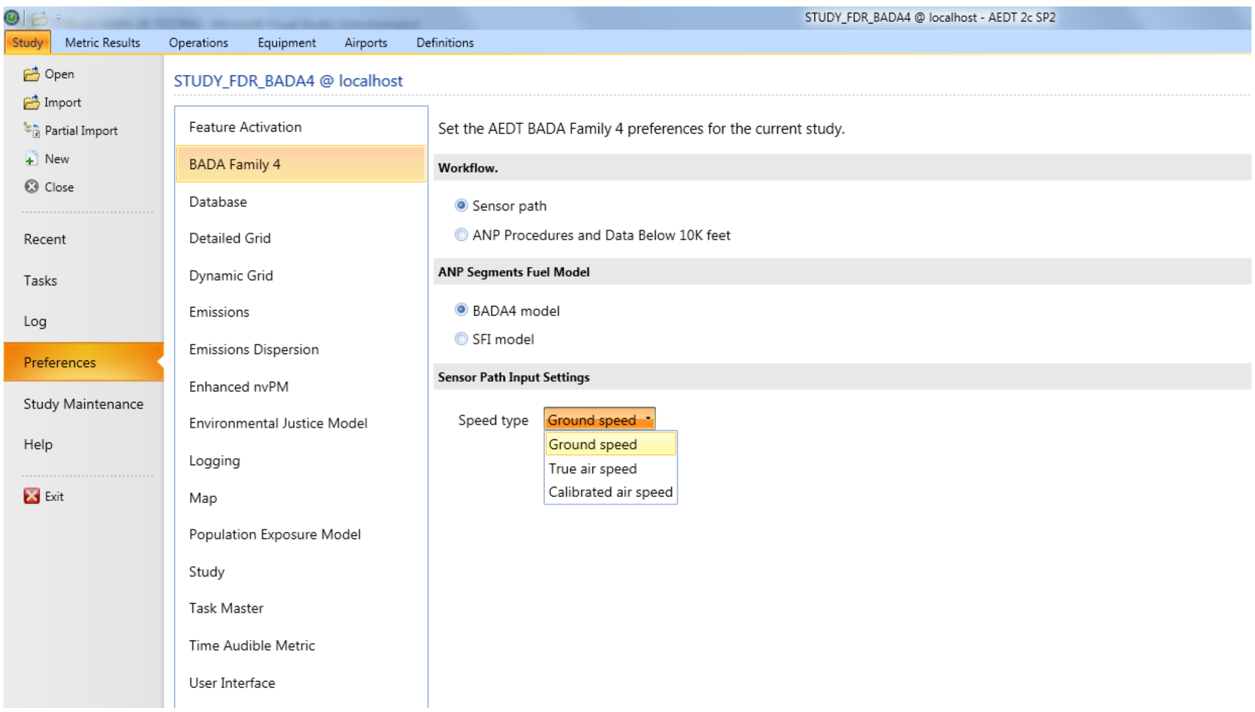


Figure 4. Screen shot of AEDT’s updated BADA4 preferences pane

3. Verification and Validation

The Volpe center has FDR data for a number of different aircraft. These datasets contain complete flight information, including fuel consumption. The Volpe Center also has access to a set of 4-D trajectory data from FAA's Performance Data and Reporting System (PDARS). This section examines the use of these datasets for verification and validation of the AEDT/BADA4 data and methods.

FDR fuel consumption results

This section discusses results of using the AEDT/BADA4 process compared to the fuel consumption data from the FDR systems for a number of datasets in Volpe's possession. In the comparison, the taxi phase is excluded from both the FDR results and the AEDT results.

FDR results for various weather usages

The set of tables below presents the sequence of improvements to the fuel consumption modeling as better data and processes are used. In each of the tables, each row represents the fuel consumption for a single flight of a large twin-aisle aircraft. These flights were chosen because detailed time and date information in the FDR records allowed the modelers to assign the appropriate MERRA-2 data to the flight[7]. In all cases, taxi information was removed from the FDR datasets to provide a direct comparison to the AEDT/BADA4 data and methods.

In Table 2 below, the simplest and most readily available data are used to illustrate what a typical user with minimum information might encounter. The ANP weights are used for the Takeoff Weight (TOW), and ISA is used to define the atmospheric conditions in which the aircraft flies. The flights were modeled with no wind. The last two columns show the AEDT/BADA4 predicted fuel consumption and the differences for the FDR reported fuel consumption.

The poor match between the FDR and the AEDT/BADA4 fuel consumption is primarily due to the ANP database not having an exact airframe match for the FDR aircraft type. The ANP aircraft which was mapped to the FDR aircraft type used a significantly heavier takeoff weight for the modeled mission compared to the takeoff weight in the FDR data. The heavier weight of the AEDT/BADA4 aircraft translates to a significant fuel consumption difference. Users can expect this level of difference to represent a 'worst-case scenario': in most cases – particularly when modeling Airbus or Boeing aircraft - the ANP database has an airframe match to the BADA4 aircraft, so there is no need to use a mapped aircraft type. Note that the BADA4 fleet did include an exact match to the FDR aircraft type, so only the takeoff weights and ground roll data were taken from the ANP mapping.

[7] Most Volpe Center FDR data do not include date information, so MERRA-2 data can't be assigned.

Table 2. Twin-Aisle Fuel results, ANP weight & ISA atmosphere

Flight ID	FDR Fuel (Kg)	AEDT/BADA4 Fuel (Kg)	Difference
383662	86972	163487	88.0%
383810	89403	164269	83.7%
384001	81801	160758	96.5%
385136	83606	165863	98.4%3
385698	87561	156873	79.2%

In Table 3 below, the ANP weights have been replaced by the actual reported TOW for the particular aircraft. Note that only the starting weights are used; from that initial condition, the AEDT/BADA4 model updates the weights via fuel consumption as the aircraft flies through the given trajectory. The last two columns show a significant improvement in the predicted fuel consumption compared to the ANP weights.

The results of using the actual TOW of each flight produces a significant improvement over using the ANP TOW, but the fuel consumption differences are still large. Again, this may be close to a worst-case scenario, since these flights were modeled with no wind and ISA atmosphere, while the actual flights experienced tail-wind components on the order of 100 knots for portions of the en-route cruise, with some flights having tail-winds close to 150 knots. When no winds are used in the AEDT/BADA4 methods, the air-mass is modeled as stationary, so the the ground speeds and the true airspeed are equal. In this case, the result is an unrealistically high true airspeed in AEDT/BADA4, which results in an unrealistically high AEDT/BADA4 fuel consumption.

Table 3. Twin Aisle Fuel results, actual TOW & ISA atmosphere

Flight ID	FDR Fuel (Kg)	AEDT/BADA4 Fuel (Kg)	Difference
383662	86972	109681	26.1%
383810	89403	110821	24.0%
384001	81801	107441	31.3%
385136	83606	104441	24.9%
385698	87561	105029	19.9%

In Table 4 below, the actual TOW are again used, and the ISA atmosphere has been replaced by the use of the associated MERRA-2 weather. We note that a separate analysis showed that improvements in the fuel consumption due to using detailed weather are equally attributable to winds and to temperature. Winds are important because the AEDT/BADA4 process models the physics of the aircraft moving through the actual air-mass. The temperatures are important because these data are needed to correctly model the Mach number of the aircraft – Mach number in turn is a key parameter in the BADA4 fuel flow calculations.

With the use of the actual TOW and the MERRA-2 winds and temperature information, the fuel consumption results are with the range of the expected accuracy of the BADA4 model. The exception is flight 384001, which still shows a large difference between the FDR and the AEDT/BADA4 fuel consumption – this flight is discussed below. Note that the average difference from the FDR fuel using ISA weather is above 25%; the average difference using MERRA-2 is less than 3%.

Table 4. Twin Aisle Fuel results, actual TOW & MERRA-2 weather

Flight ID	FDR Fuel (Kg)	AEDT/BADA4 Fuel (Kg)	Difference
383662	86972	88280	1.5%
383810	89403	89180	-0.2%
384001	81801	87577	7.1%
385136	83606	85208	1.9%
385698	87561	89274	2.0%

Finally, Table 5 below shows the results of combining the detailed weight and weather information with the smoothing operations discussed earlier in this document. We note that flight 384001 in this dataset experiences tail-winds on the order of 150 knots; and in the cruise phase of flight the true airspeed varied on the order of ten knots between consecutive FDR readings. Without smoothing, the AEDT/BADA4 tries to following these airspeed deviations; with smoothing, these deviations are minimized.

Table 5. Twin Aisle Fuel results, actual TOW & MERRA-2 weather, smoothing

Flight ID	FDR Fuel (Kg)	AEDT/BADA4 Fuel (Kg)	Difference
383662	86972	87030	0.1%
383810	89403	88742	-0.7%
384001	81801	83956	2.6%
385136	83606	83218	-0.5%
385698	87561	87368	-0.2%

FDR results for various smoothing/filtering usages

Tests with the FDR data showed only minor improvements for total flight fuel consumption for the five flights under close consideration. The majority of the fuel consumption occurs in cruise, where the aircraft does not significantly change and so the effects of smoothing and filtering are minimal[8].

Smoothing and filtering can have its largest effects on terminal area operations, where airspeeds and altitude are changing relatively quickly. Examples of the effects of different methods of smoothing and filtering on FDR Flight #384001 are shown in Figure 5 and Figure 7 below.

Figure 5 presents the fuel flow data from the FDR system, as well as from AEDT/BADA4 with and without smoothing. The instantaneous fuel flow during a sample departure as recorded by the FDR is compared with values modeled in AEDT using unaltered FDR trajectory (legend key 'AEDT') and using smoothed trajectory data (legend key 'AEDT with smoothing'). No filtering is applied in this example and speeds

[8] Minimal, but not zero; Smoothing and filtering did remove some en-route airspeed anomalies on one of the flights.

are not re-calculated from smoothed ground track data during smoothing. Fuel flow data from the smoothed trajectory case show fewer high frequency fluctuations compared with the input AEDT case, while still following trends of the FDR data.

Figure 6 continues the comparison – begun in Figure 5 – by illustrating the cumulative fuel consumption. Altitude information is also presented in the graphic so the reader can see why the fuel flow reduces significantly between 10 and 15 nautical miles from brake-release: the aircraft had an altitude hold at 5000 feet in this region. With smoothing, but with no filtering, the AEDT fuel flow data show the effects of trying to follow small changes in the speed and climb gradients – the AEDT calculated fuel flow is significantly noisier than the actual fuel flow data.

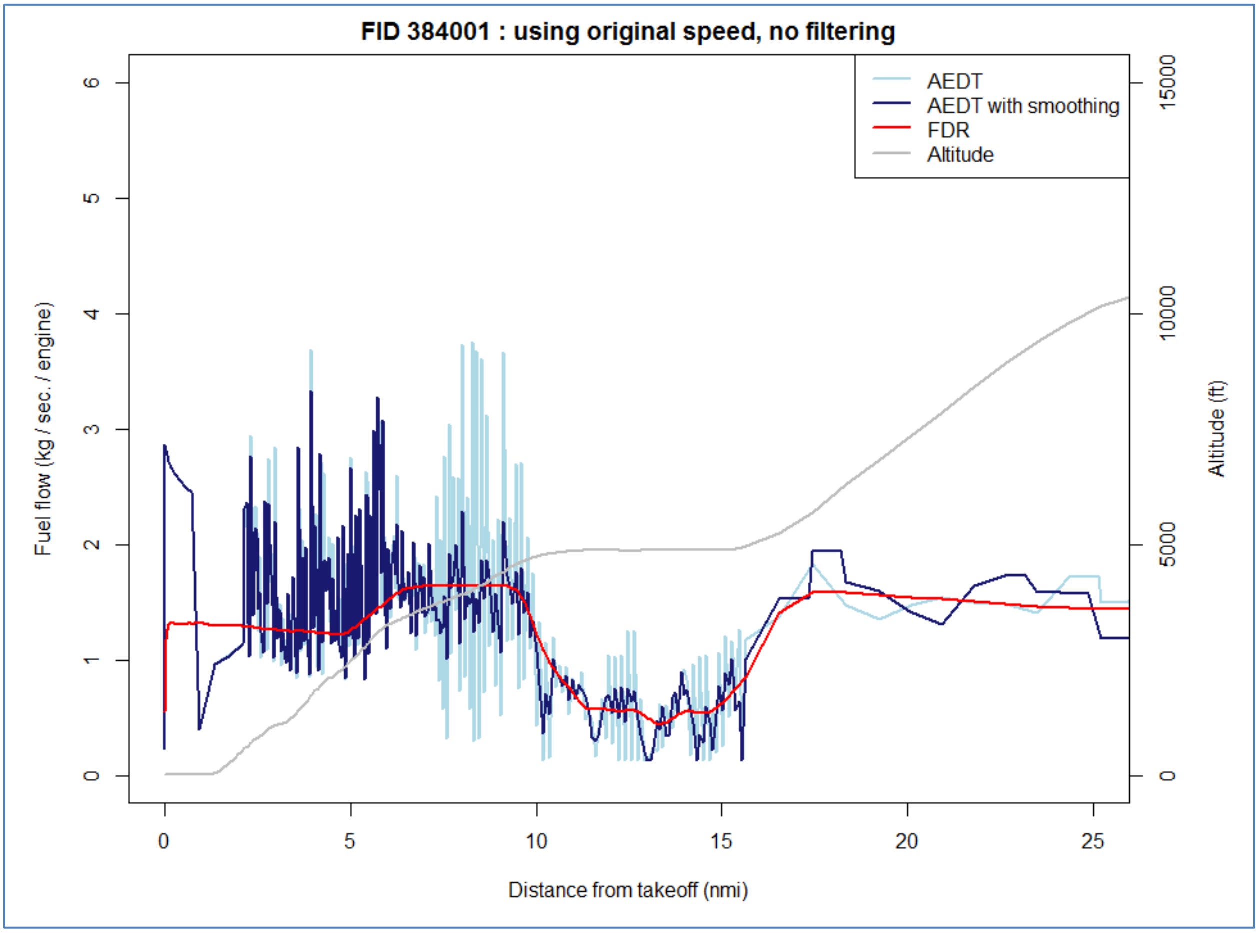


Figure 5. Instantaneous fuel flow for a sample departure with no filtering

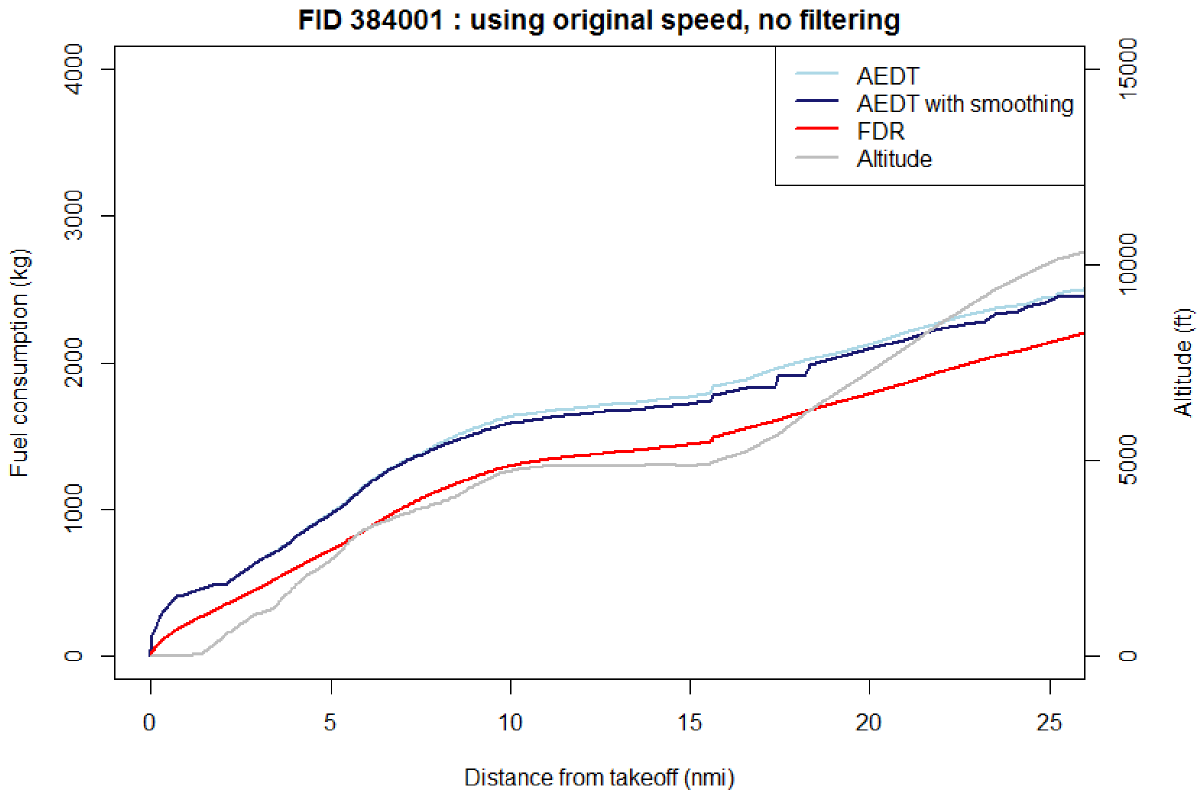


Figure 6. Cumulative fuel consumption for a sample departure with no filtering

In Figure 7, filtering has been added to the process. The general over-prediction in fuel flow between 3 and 7 nautical miles is still present, but the rapid, unrealistic fluctuations in the fuel flow are significantly reduced – though they are not eliminated. Figure 8 illustrates the cumulative fuel consumption for the case with filtering; for this particular flight, after 25 nautical miles, the net effect of filtering on fuel consumption is a slightly closer prediction to the FDR (as compared to the no-filtering case in Figure 6). This improvement in the prediction due to filtering is not necessarily extensible to all flights in general.

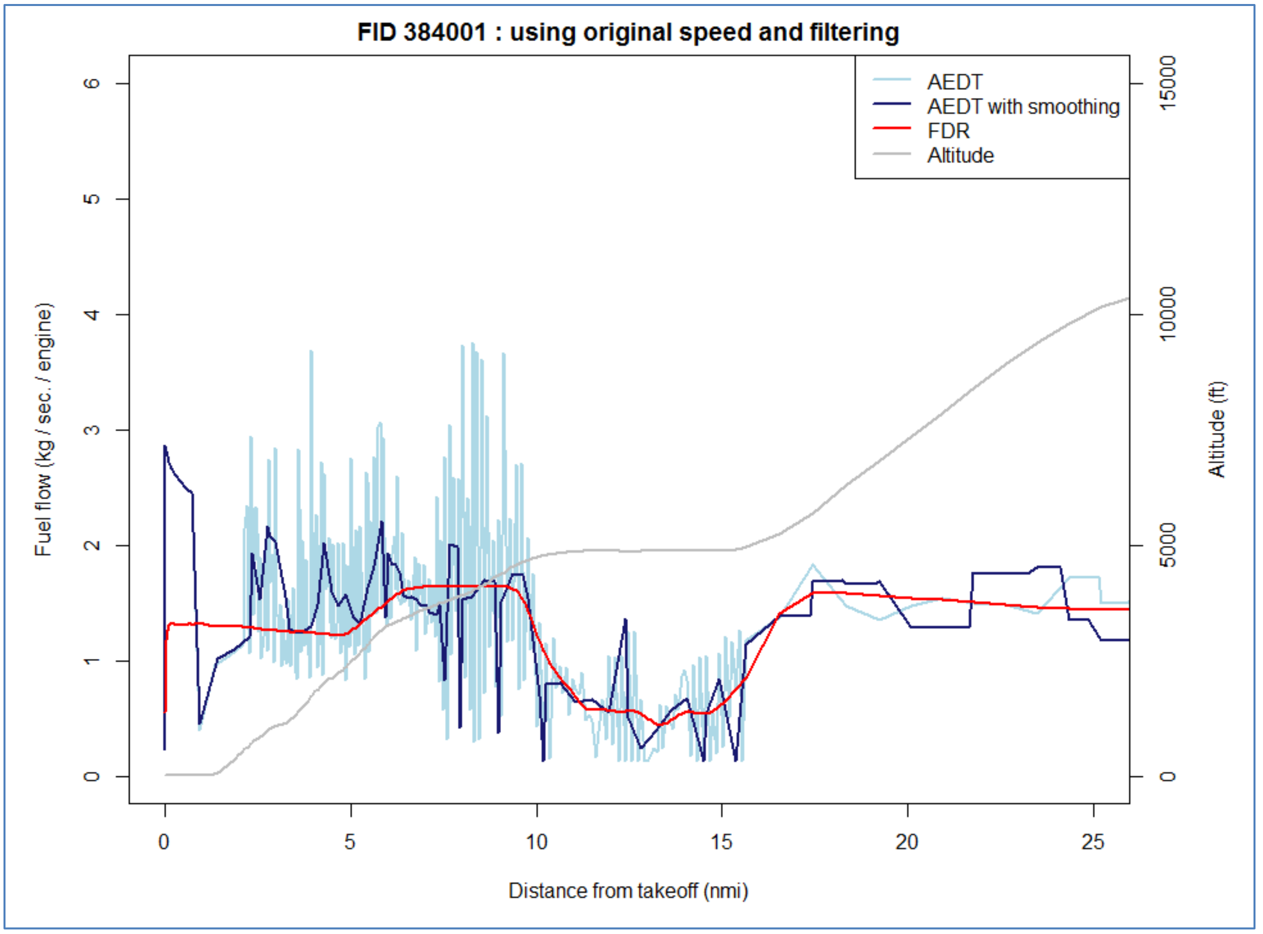


Figure 7. Instantaneous fuel flow for a sample departure with filtering

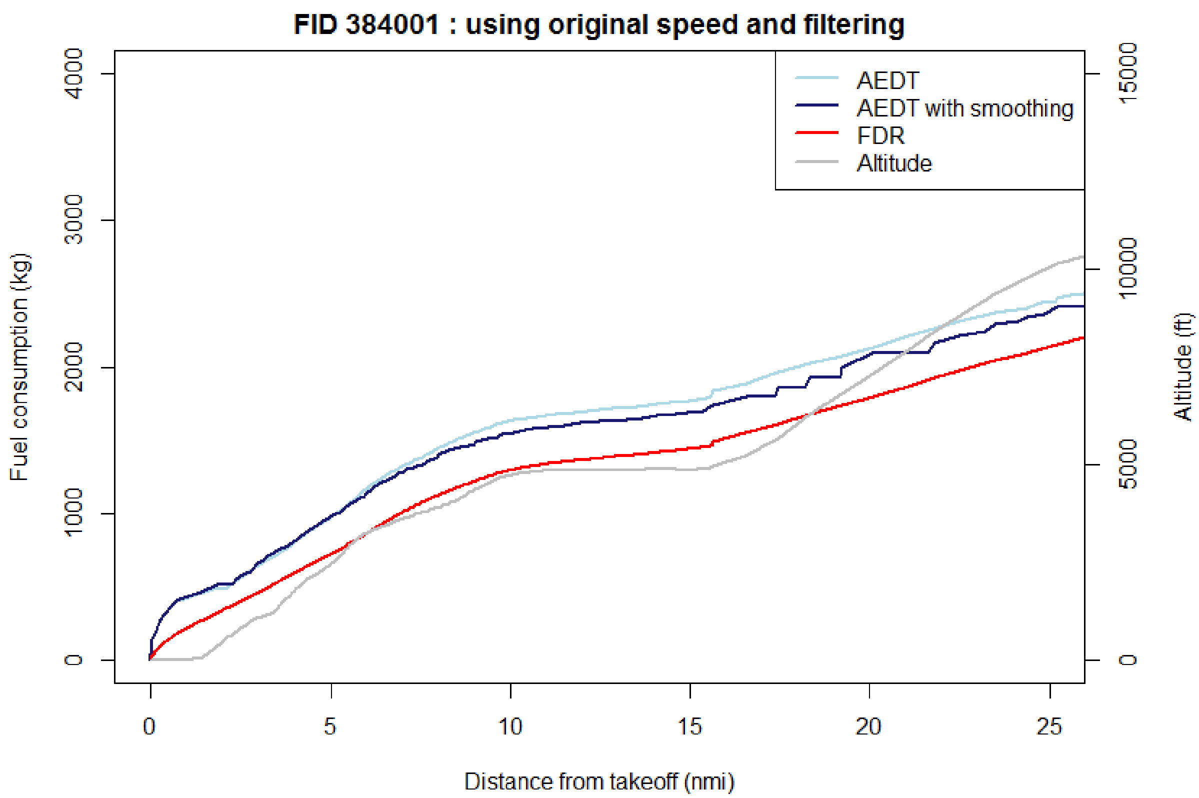


Figure 8. Cumulative fuel consumption for a sample departure with filtering

Comparison against FDR where weights and weather data are not available

One of the datasets is a single aisle aircraft for an airline based in Europe. We used this dataset to compare the AEDT/BADA4 process for an example of when detailed weight and weather are not or cannot be used. The comparison is given in Figure 9. Note that the FDR dataset used in this section is not the dataset used above.

In Figure 9, each datum represents the fuel consumption for one of the 178 flights in this dataset. The location of each datum along the horizontal axis represents the 'truth' – what the FDR system reported as the fuel consumption for the flight. The location of each datum relative to the vertical axis represents the AEDT modeled fuel consumption of that particular flight. The figure also includes a line labeled "Perfect Fit" – this is the line where the values along the horizontal axis equal the values along the vertical axis. If a data point falls on this line, the fuel consumption predicted by AEDT match the fuel consumption reported by the FDR system. Values above the "Perfect Fit" line represent an over-prediction of the fuel consumption by AEDT; values below the line represent an AEDT under-prediction.

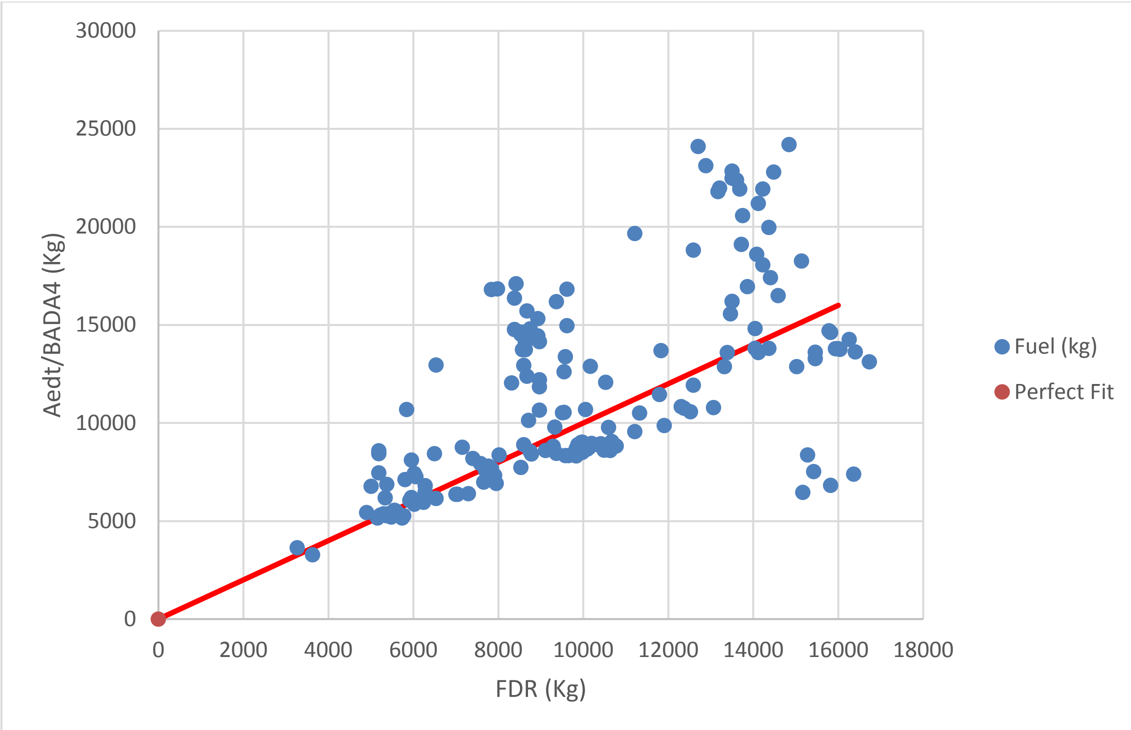


Figure 9. Comparison of FDR and AEDT/BADA4 for a single aisle aircraft - ANP weights & ISA weather

[space intentionally blank]

To give a sense of the magnitude of these differences, Figure 10 below shows the flight distance and the fuel consumed as reported by the FDR system for these flights. For longer flight distances (between 3,000 and 3,500 nautical miles), the fuel consumption can vary by on the order of 4000 Kg, or about 25% for flights by the same aircraft. So for a given aircraft type flying a given distance, the fuel consumption can vary on the order of 25% due to the variation in takeoff weights and en-route atmospheric conditions.

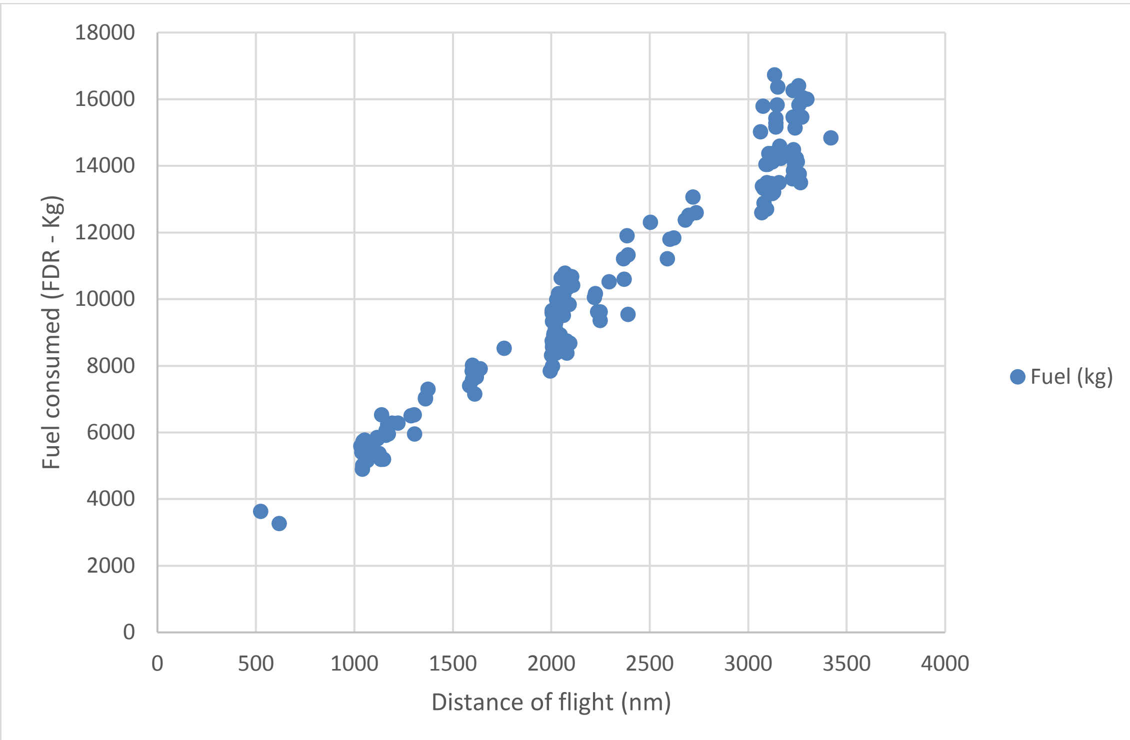


Figure 10. Fuel consumed by flight distance for the flights in Figure 9

PDARS fuel consumption results

PDARS study sample results

Figure 11 compares full-flight results of AEDT-BADA4 to a prior AEE inventory generated with AEDT-BADA3/ANP data. The scope of the comparison is a three-day period of departures from one major U.S. airport, with each data point in the graphic representing a given combination of route (origin and destination) and aircraft/equipment. The figure shows that:

- On average, the observed difference in fuel per unit distance is lower in BADA4 by 1.7%
- Individual points, i.e., route-and-equipment pairs, typically are within +/-10%
- Some outliers are known to be due to a difference in the TOW assumption

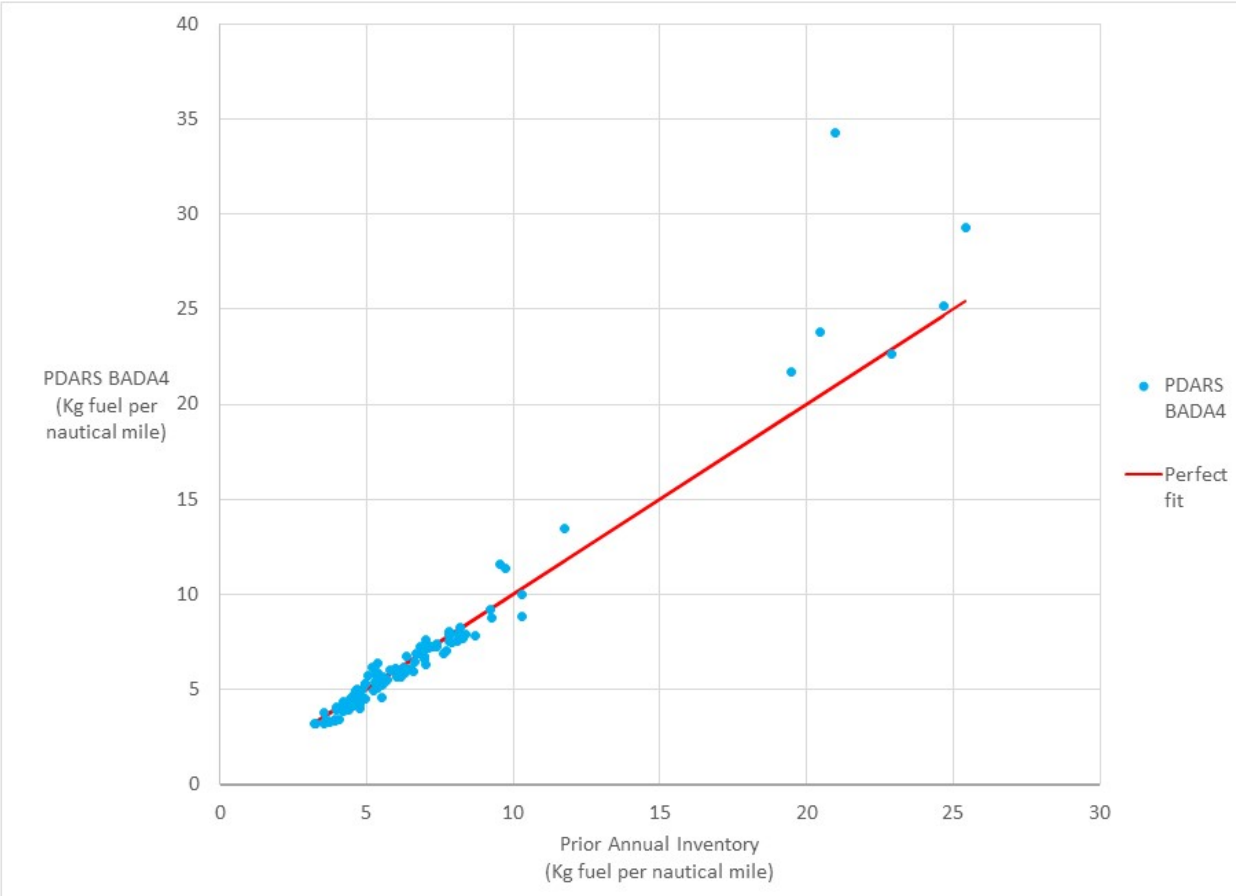


Figure 11. BADA4 comparison against prior inventory for departures from a U.S. airport

MERRA-2 Weather

A validation of the MERRA-2 weather was conducted by comparing the FDR-reported wind information to the data for the same 4-D position from the MERRA-2 data. The FDR data are from the five flights discussed in Section 0. The standard deviation for the temperature was less than one degree Celsius. For two of the flights, the standard deviation of the wind magnitude was less than five knots. Three of the flights contained wind magnitude data drop-outs in the FDR data. Even with these data drop-outs, the wind magnitude standard deviation was on the order of ten knots. The local pressure was also compared; the MERRA-2 values had a standard deviation of 2 or 3 millibars from the FDR data.

4. Guidance for using the BADA4 Sensor Path Implementation in AEDT

This section discusses some of the potential issues with using the BADA4 version of AEDT of which the user should be aware.

For additional, general guidance on setting up BADA4 studies, see the following appendices:

- Appendix A: BADA4 Workflow Limitations and Potential Work-arounds
- Appendix B: Installing FLEET Database with BADA4 Data

Importance of smoothing and filtering

Testing has demonstrated that raw, rapid-update, and generally noisy radar trajectories are required to be pre-processed – i.e., adjusted outside of existing AEDT methods using tools like the SPI Tool – in order for the BADA4 Sensor Path Implementation to produce reasonable fuel consumption results.

Naturally, while the smoothing and filtering pre-processor may help to eliminate issues due to poor input data quality, the user should be aware that information in the original dataset will be lost when smoothing and filtering is applied. If the user is looking for the effects of small changes in procedures in noisy trajectory data, the use of smoothing and filtering should be done cautiously to ensure that the procedure changes are not removed by the pre-processing. The smoothing and filtering pre-processor is intended to be user-configurable and as such, a set of eight parameters can be used to control the underlying algorithms. Table 1 above lists those parameters and the internal validation rules. Those rules are intended to prevent pre-processor failures, and the user should be aware that certain parameter values may allow the pre-processor to run without problems but still produce values which are not meaningful. The default values listed in Table 1 provide some guidance for choosing reasonable parameter values. What constitutes a reasonable value of some parameters, the window widths in particular, will depend on the resolution of the input data.
For additional guidance regarding smoothing and filtering, see *Appendix C: Sensor Path Import Tool*.

Users still have the option of using AEDT's existing smoothing and filtering methods, which are described in section 3.7.2.1 of the AEDT Technical Manual. Note that the existing methods are integrated into AEDT, while the Sensor Path Import Tool is an external pre-processor.

Importance of takeoff weights

Aircraft weight is a primary influence of aircraft performance. Using incorrect weights will result in incorrect fuel consumption values. That said, there may be studies where the differences in the base case and the alternative cases are more important than the absolute fuel consumption numbers. In this case, a consistent use of weights across the cases will probably produce meaningful differences in aircraft fuel consumption. Users should be cognizant of using aircraft weights which match the level of detail required for the study.

Using high fidelity weather

Testing has determined that the existing BADA3 process will not work with MERRA-2 data which contains 2-D (surface weather). We have noted no issues with BADA3 using 3D MERRA-2 data. If the user has a study with a large percentage of non-BADA4 aircraft in their study fleet, removing the 2D surface weather may be a better option than having large numbers of flights fail to process. Note that removing the 2-D weather means that AEDT will derive the surface weather from extrapolating the 3-D weather above the surface. The accuracy of this extrapolation will depend on the distance of the extrapolation, the surface topography, and the characteristics of the 3-D weather in the pressure layers used for the extrapolation.

For additional guidance, see *Appendix D: Acquiring and Preparing MERRA-2 Weather Data for AEDT Modeling*.

Importance of Fleet mappings

The BADA4 fleet contains good coverage for some of the manufacturer's fleets, but not for all. Coverage is good for in-production Airbus, Boeing, and Embraer aircraft, but other manufacturers are not represented[9]. Users should note that they do have the ability to create user-defined aircraft if they believe that a mapping from a particular airframe and engine combination currently found in AEDT is not representative of how they wish to model aircraft in their study. Ultimately, the fuel consumption results are only as good as the choice of aircraft used for the study.

For convenience, a default mapping table of ICAO aircraft types (to AEDT equipment) is provided and available for use in the Sensor Path Import (SPI) Tool. Note that not all ICAO aircraft types are covered and that the table is available in the Fleet database (see 'FltActypeToUniqueEquipMap'). *Appendix E: Creation of a Default ICAO Type Mapping Table* contains the simple SQL query used to populate the default mapping table; here the user will note that engine choice is arbitrary (among the available AEDT engines tied to the particular airframe of interest). Alternatively, when ingesting their operations using SPI, the user has the ability to explicitly define the operation via the equipment ID.

---

[9] BADA4 contains some Fokker aircraft, but these types are not significant in the U.S. fleet. BADA4 also contains a smaller number of turboprops and piston engine aircraft, but these are not modeled with BADA4 in AEDT.

BADA3 usage

As implied in the prior section, not all aircraft in the fleet are modeled with BADA4 aircraft. AEDT will automatically revert to using a BADA3 aircraft if a BADA4 aircraft is not available, or if the BADA4 aircraft fails to process correctly in the AEDT/BADA4. That is, rather than simply failing the flight if the BADA4 process fails, AEDT will next attempt to process the flight with the BADA3 methods. Note that the BADA3 methods also use the MERRA-2 high fidelity weather, when available, with the exception that 2D surface data are not used in the BADA3 methods. In the event a BADA4 aircraft fails, this operation is reported in the log files.

Pressure and Geometric Altitude

The BADA4 process currently assumes that the input altitude is the geometric altitude of the aircraft in feet MSL. Most sensor path data are pressure altitude. Testing showed that the difference between using pressure altitude and geometric altitude resulted in fuel consumption differences on the order of one percent.

Flaps and slats issues

The AEDT/BADA4 process uses the BADA4 flaps schedules. Testing has shown that the AEDT/BADA4 process uses a flap deployment schedule on approach that is significantly different than that in the FDR data. The AEDT/BADA4 process has the first flap and slat deployment occur at 8000 feet AGL. The actual flap deployment in the FDR dataset was typically much lower – initial slats and flaps were typically deployed at less than 3000 feet AGL. This translates to the AEDT/BADA4 process over-estimating the drag of the aircraft below 8000 feet. We recommend not using the current version of AEDT/BADA4 for studies where procedures changes occur below 8000 feet AGL. The AEDT development team plans to incorporate the ANP arrival flaps schedule for the next version of AEDT; the use of the ANP flaps schedule should lower the recommended altitude to significantly below 6000 feet AGL.

5. References

A-21 Committee. (1986). *Procedure for the calculation of airplane noise in the vicinity of airports.* Warrendale, PA: SAE.

Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica, 10*(2), 112-122.

EUROCONTROL Experimental Centre. (2014). *User manual for the Base of Aircraft Data (BADA) revision 3.12.* Sur de Orge: European Organisation for the Safety of Air Navigation.

Federal Aviation Administration. (2016). *Aviation Environmental Design Tool (AEDT) Technical Manual, Version 2c.* Washington, DC: Office of Environment and Energy.

Pearson, R. K., Neuvo, Y., Astola, J., & Gabbouj, M. (2016). Generalized Hampel Filters. *EURASIP J. Adv. Signal Process.*, 87.

Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing, 1*(3), 244-256.

Senzig, D. A., Fleming, G. G., & Iovinelli, R. J. (2009, July-August). Modeling of Terminal-Area Airplane Fuel Consumption. *Journal of Aircraft, 46*(4), 1089-1093.

Appendix A: BADA4 Workflow Limitations and Potential Work-arounds

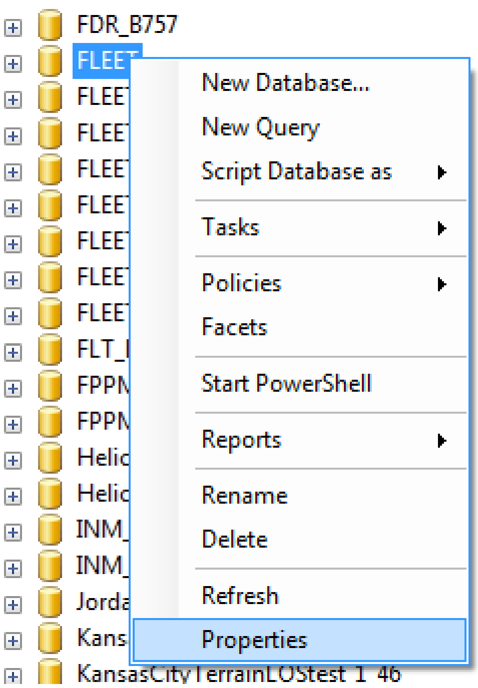
1. To enable BADA4 modeling in AEDT, a user needs to enter the appropriate identifier and hash key on the Feature Activation pane and make sure that the modeled Study contains the BADA4 data.
2. The FLEET database and the standard sample Studies provided with the public AEDT install do not contain BADA4 data.
3. Activation of the BADA4 modeling through the Feature Activation does not bring BADA4 data into the user's Study.
4. The user must replace the standard FLEET database with the public AEDT install with the FLEET_BADA4 database provided by Volpe on request.
5. The replacement of the standard AEDT FLEET with the FLEET_BADA4, does not have an immediate effect on the user's existing Studies.
6. If the user creates a new Study, it will have BADA4 data.
7. If the user has an existing Study v.1.53.1 without BADA4 data, the user could use a SQL script included below to inject BADA4 data into the existing Study. The user needs to make sure that item #: 4 above is complete, copy the content of the script to a new query window of SQL Server Management Studio (SSMS), in the "use" replace the "STUDY_NAME" with the real Study name, and execute the script.
8. If the user has an existing Study v.1.53.0 without BADA4 data, the AEDT Study upgrade feature will result in the Study v.1.53.1 with BADA4 data.
9. If the user has an existing Study older than v.1.53.0 without BADA4 data, the AEDT Study upgrade feature will result in the Study v.1.53.1 without BADA4 data. An advanced user can inject BADA4 data into the Study, see item #5.
10. Current BADA4 implementation supports modeling of the full flight, partial departure, and partial arrival runway to runway sensor path operations. The Study's SENSOR_PATH table should contain either the full runway to runway trajectories, or partial arrival or departure trajectories of the modeled flights. This branch of the BADA4 workflow corresponds to the "Sensor path" workflow option of the "BADA Family 4" preferences.
11. BADA4 workflow can also model so called track based runway to runway operations when the Study's APT_SEGMENT table contains the great circle trajectories of the modeled flights. The public AEDT install provides an example of this type of the Study, STUDY_IFSET. This branch of the workflow is also known as CAEP workflow, and corresponds to the "ANP procedures and DATA below 10K" workflow option of the "BADA Family 4" preferences.
12. The BADA4 workflow does not support modeling of the terminal ANP profile based operations. For example, standard AEDT sample studies, STUDY_INM, STUDY_NIRS, and STUDY_DULLES cannot be processed by the BADA4 workflow.

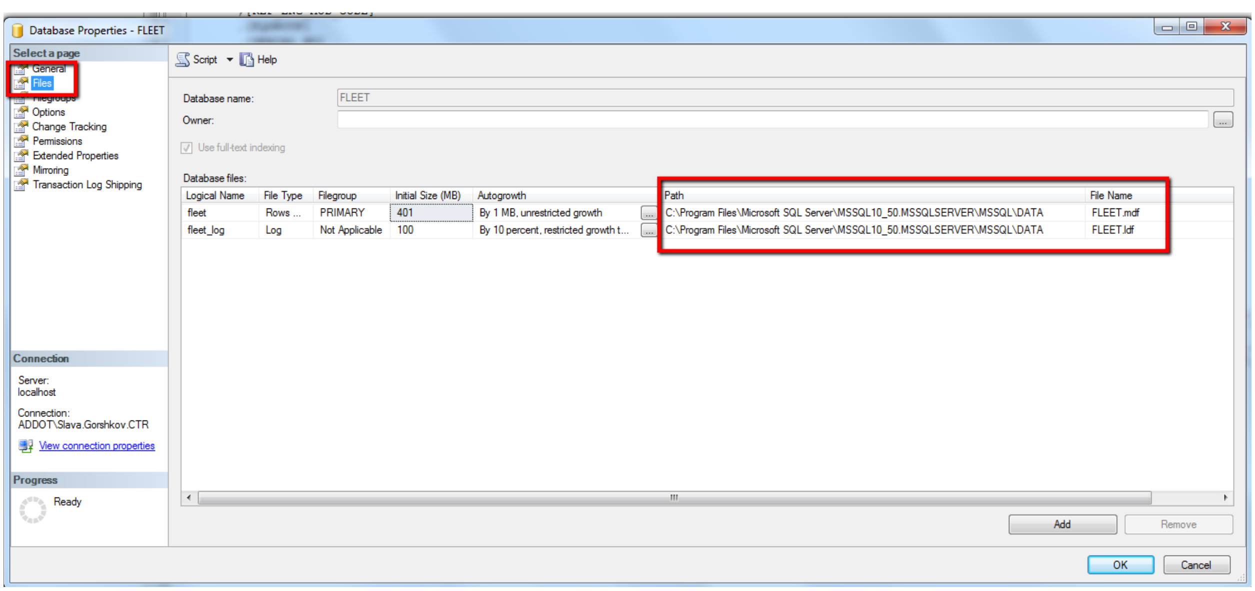
13. Current BADA4 implementation can only work with the ANP procedures. If a FLEET database equipment contains both ANP profiles and ANP procedures, according to the AEDT business logic, the respective AEDT aircraft will contain only the ANP profile points. In addition, certain FLEET equipment contains only the ANP profiles points. A SQL script could be provided to an advanced user to make this equipment usable in the BADA4 workflow. However, this action will make the equipment different from the standard FLEET database's equipment.

Appendix B: Installing FLEET Database with BADA4 Data

This document describes the steps to replace the standard AEDT FLEET database with the FLEET with BADA4 database.

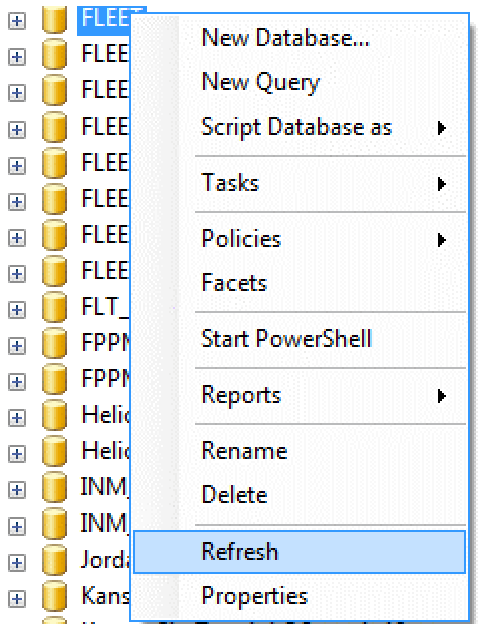
1. Open SQL Server Management Studio (SSMS). Verify and record the current names and locations of the FLEET data and log files by following the steps below:



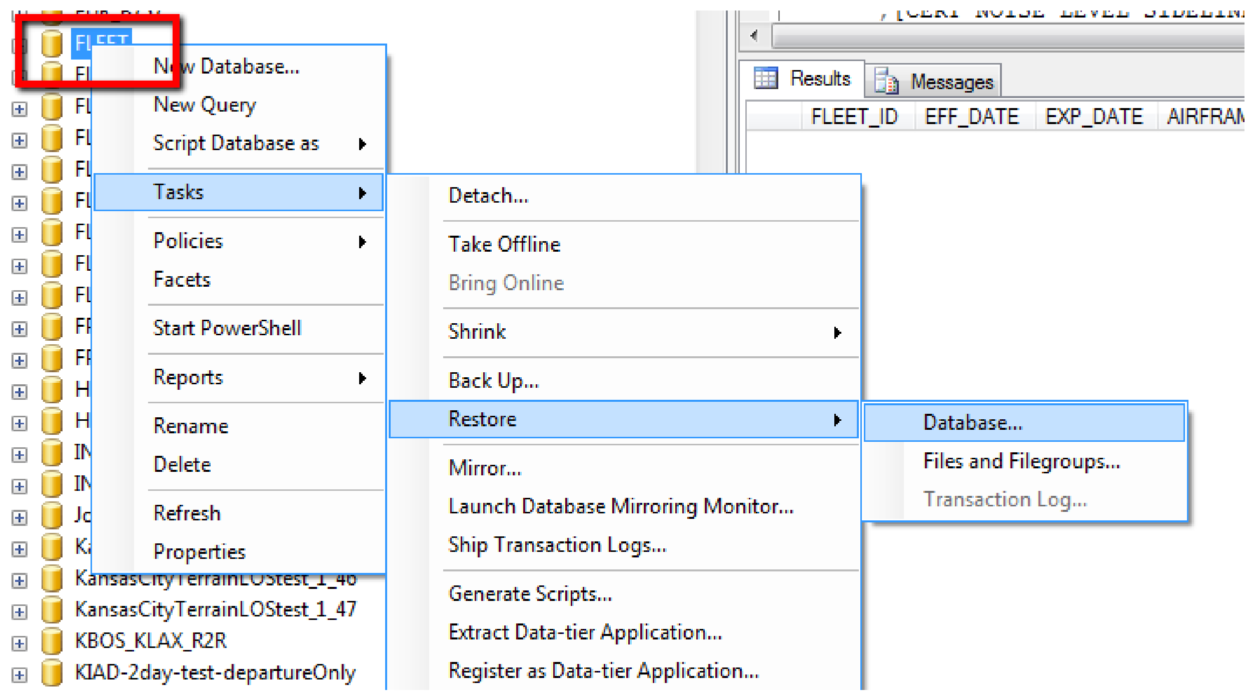


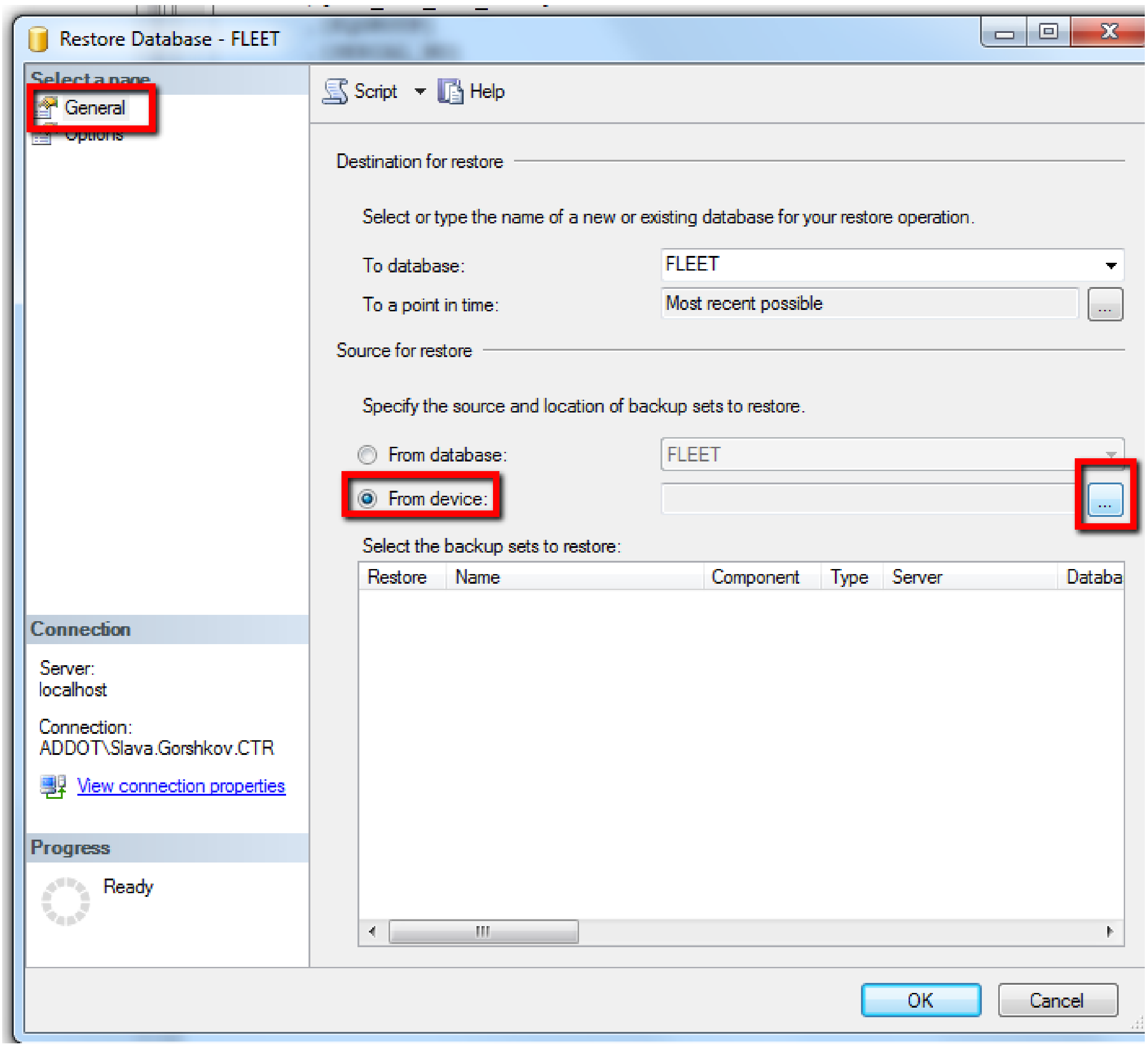
2. Close AEDT application if it is open.

3. Make sure FLEET_BADA4.bak is available and accessible by the user.
4. **If the user has SSMS 2008**.
   4.1. Close all open queries in SSMS.
   4.2. Refresh the FLEET database:

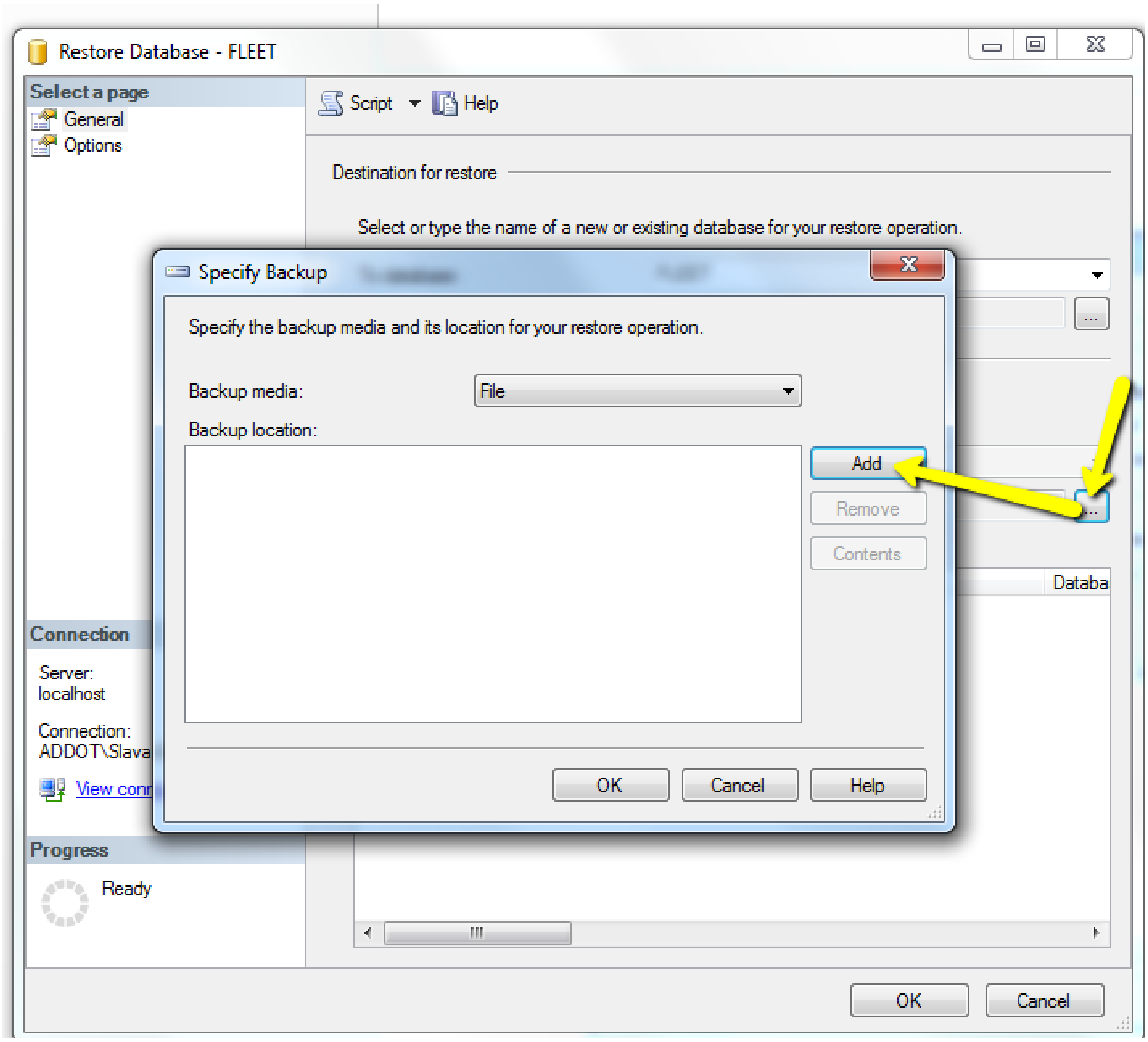


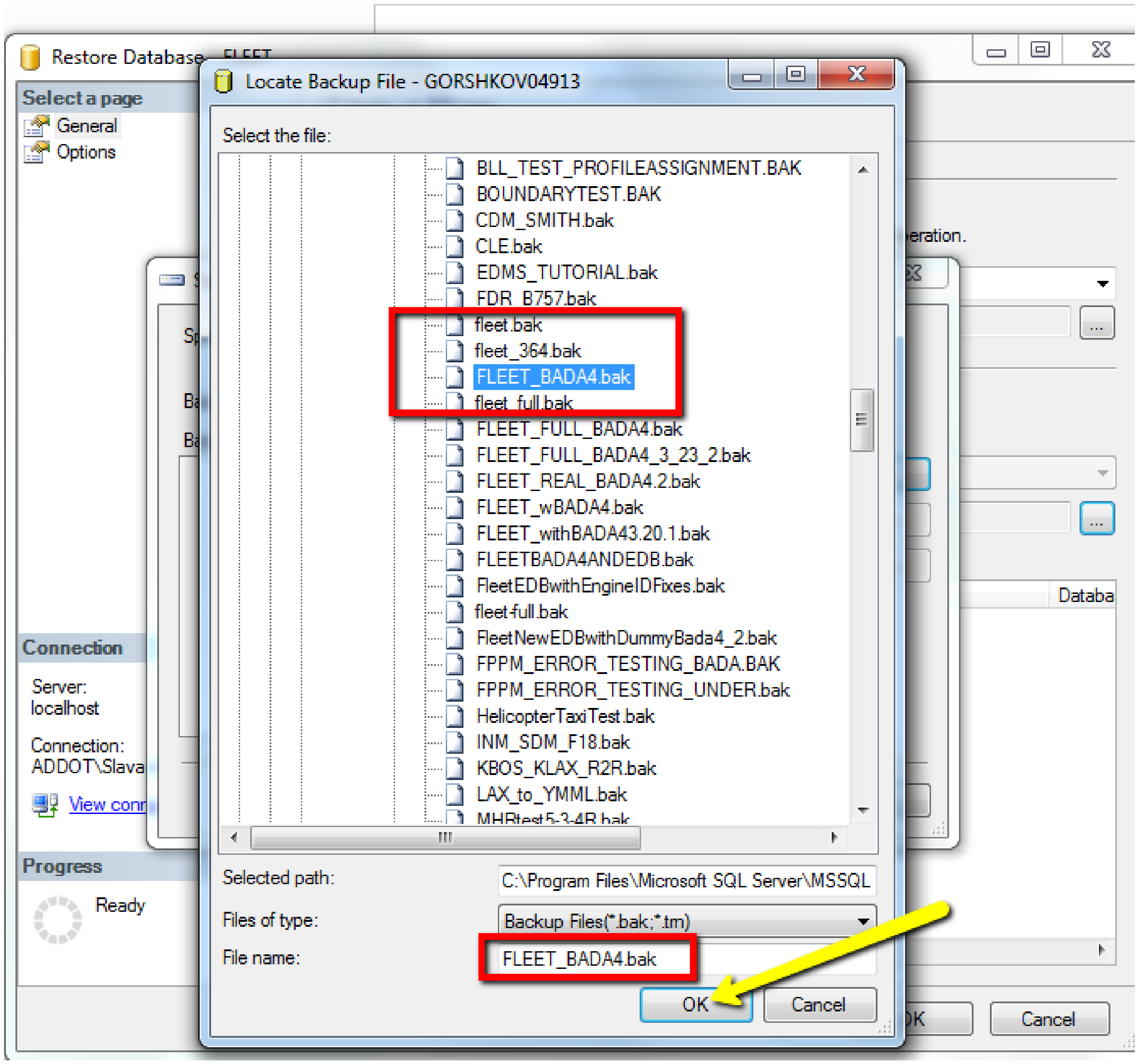
   4.3. Restore the FLEET_BADA4 as FLEET:

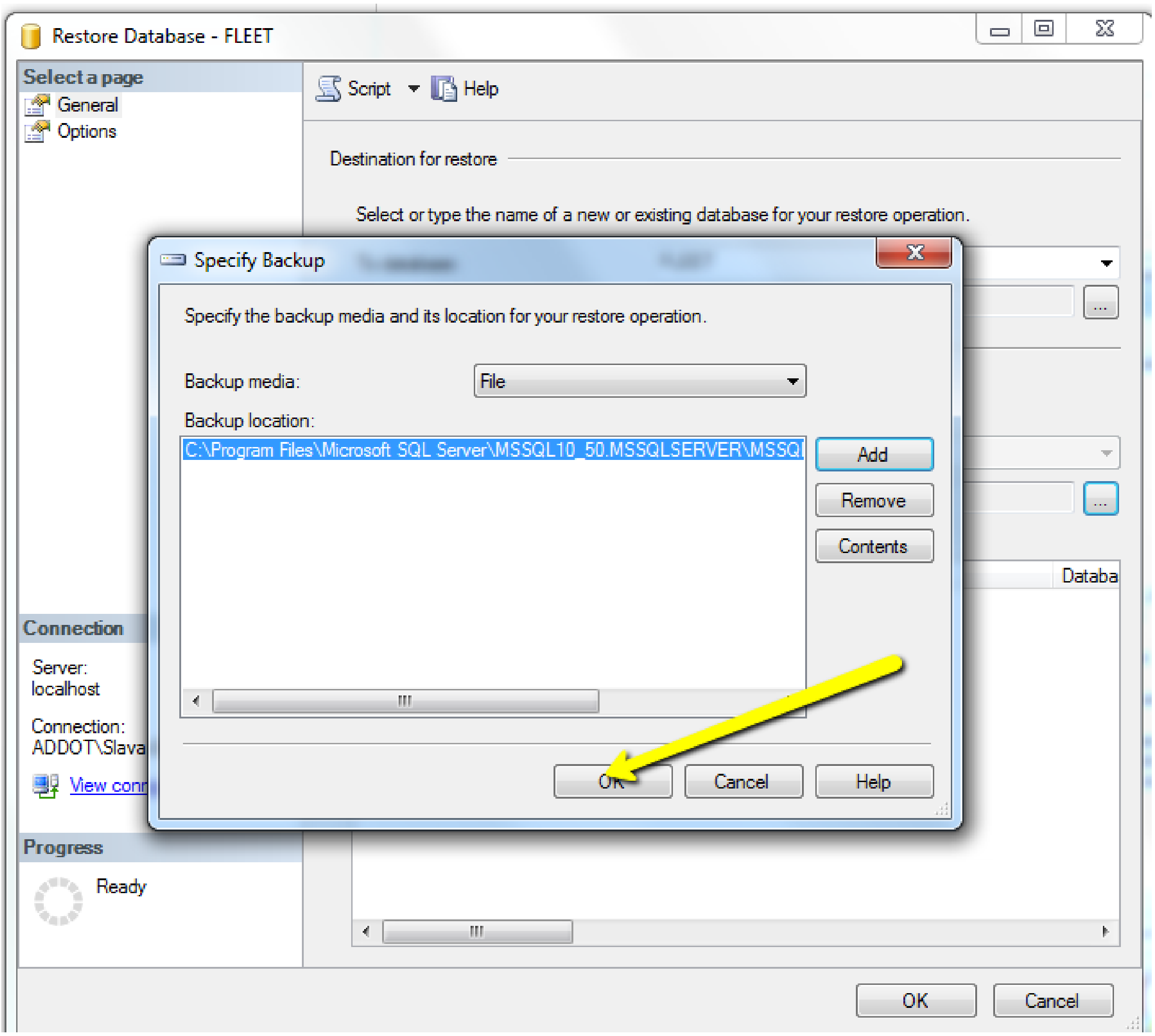


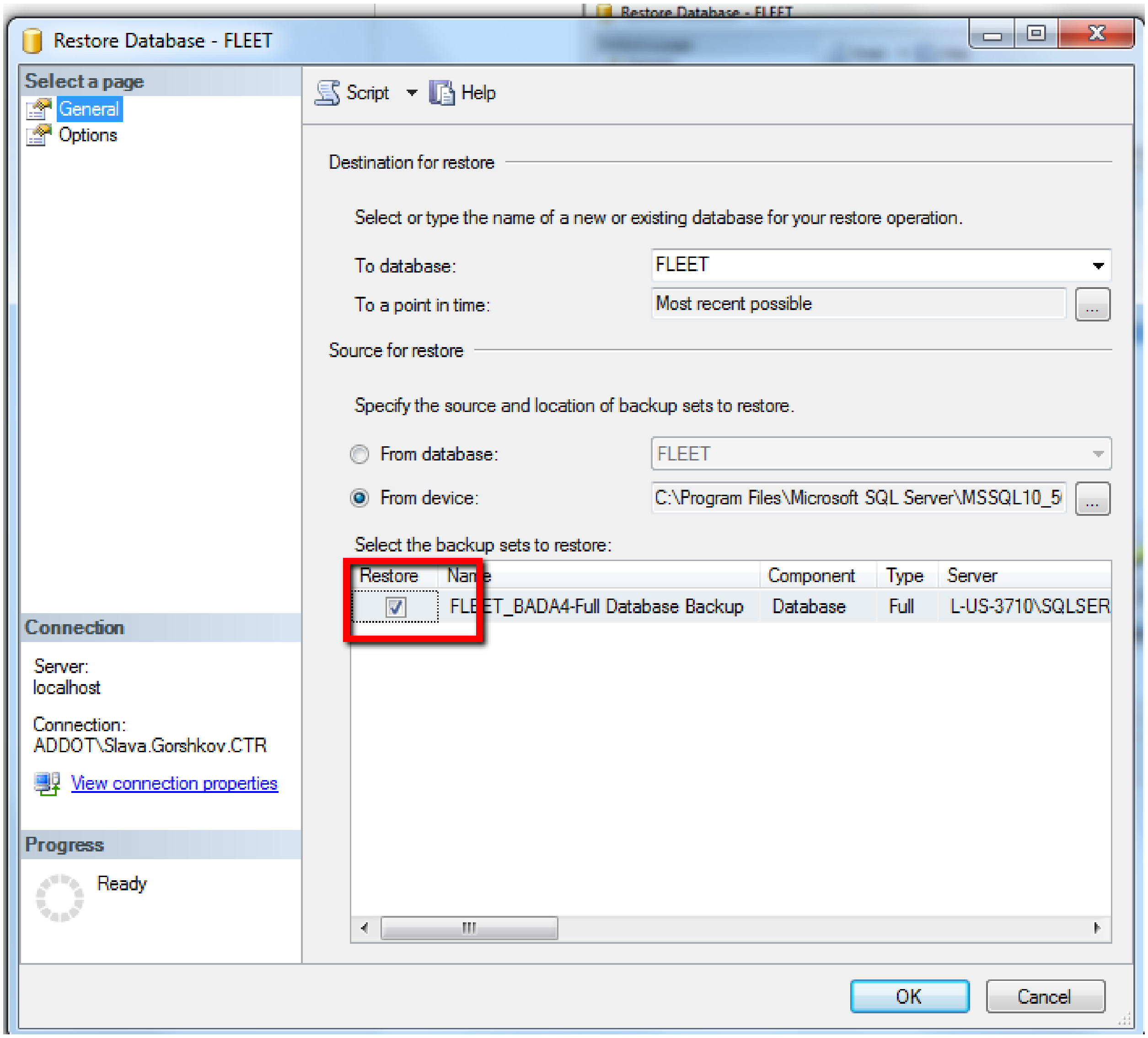


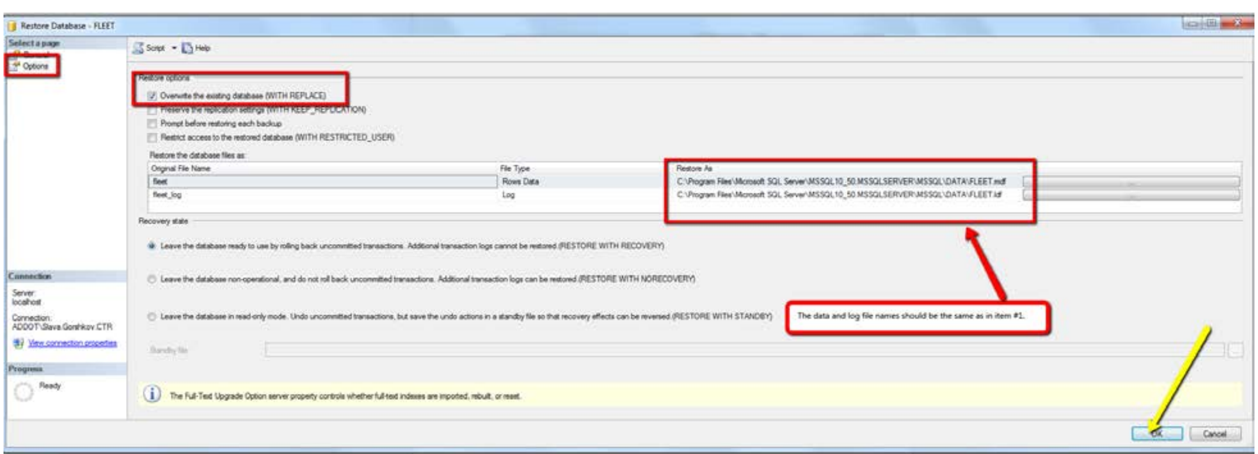
Restore Database - FLEET
Select a page
General
Options
Script
Help
Destination for restore
Select or type the name of a new or existing database for your restore operation.
To database:
FLEET
To a point in time:
Most recent possible
Source for restore
Specify the source and location of backup sets to restore.
From database:
FLEET
From device:
Select the backup sets to restore:
Restore
Name
Component
Type
Server
Databa
Connection
Server:
localhost
Connection:
ADDOT\Slava.Gorshkov.CTR
View connection properties
Progress
Ready
OK
Cancel


Restore Database - FLEET
Select a page
General
Options
Script
Help
Destination for restore
Select or type the name of a new or existing database for your restore operation.
Specify Backup
Specify the backup media and its location for your restore operation.
Backup media:
File
Backup location:
Add
Remove
Contents
OK
Cancel
Help
Databa
Connection
Server:
localhost
Connection:
ADDOT\Slava
View conn
Progress
Ready
OK
Cancel


Restore Database - FLEET
Locate Backup File - GORSHKOV04913
Select a page
General
Options
Connection
Server:
localhost
Connection:
ADDOT\Slava
Progress
Ready
Select the file:
BLL_TEST_PROFILEASSIGNMENT.BAK
BOUNDARYTEST.BAK
CDM_SMITH.bak
CLE.bak
EDMS_TUTORIAL.bak
FDR_B757.bak
fleet.bak
fleet_364.bak
FLEET_BADA4.bak
fleet_full.bak
FLEET_FULL_BADA4.bak
FLEET_FULL_BADA4_3_23_2.bak
FLEET_REAL_BADA4.2.bak
FLEET_wBADA4.bak
FLEET_withBADA43.20.1.bak
FLEETBADA4ANDEDB.bak
FleetEDBwithEngineIDFixes.bak
fleet-full.bak
FleetNewEDBwithDummyBada4_2.bak
FPPM_ERROR_TESTING_BADA.BAK
FPPM_ERROR_TESTING_UNDER.bak
HelicopterTaxiTest.bak
INM_SDM_F18.bak
KBOS_KLAX_R2R.bak
LAX_to_YMML.bak
Selected path:
C:\Program Files\Microsoft SQL Server\MSSQL
Files of type:
Backup Files(*.bak;*.tm)
File name:
FLEET_BADA4.bak
OK
Cancel


Restore Database - FLEET
Select a page
General
Options
Script
Help
Destination for restore
Select or type the name of a new or existing database for your restore operation.
Specify Backup
Specify the backup media and its location for your restore operation.
Backup media:
File
Backup location:
C:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQI
Add
Remove
Contents
OK
Cancel
Help
Connection
Server:
localhost
Connection:
ADDOT\Slava
Progress
Ready
Databa
OK
Cancel





4.4. If the Restore fails, the user could open a new query window in SSMS and issue the following commands which close the existing open connections against the FLEET database:

```
use master;

alter database FLEET
      set offline with rollback immediate
```

Then the user should try to repeat the Restore process, item # 4.3.

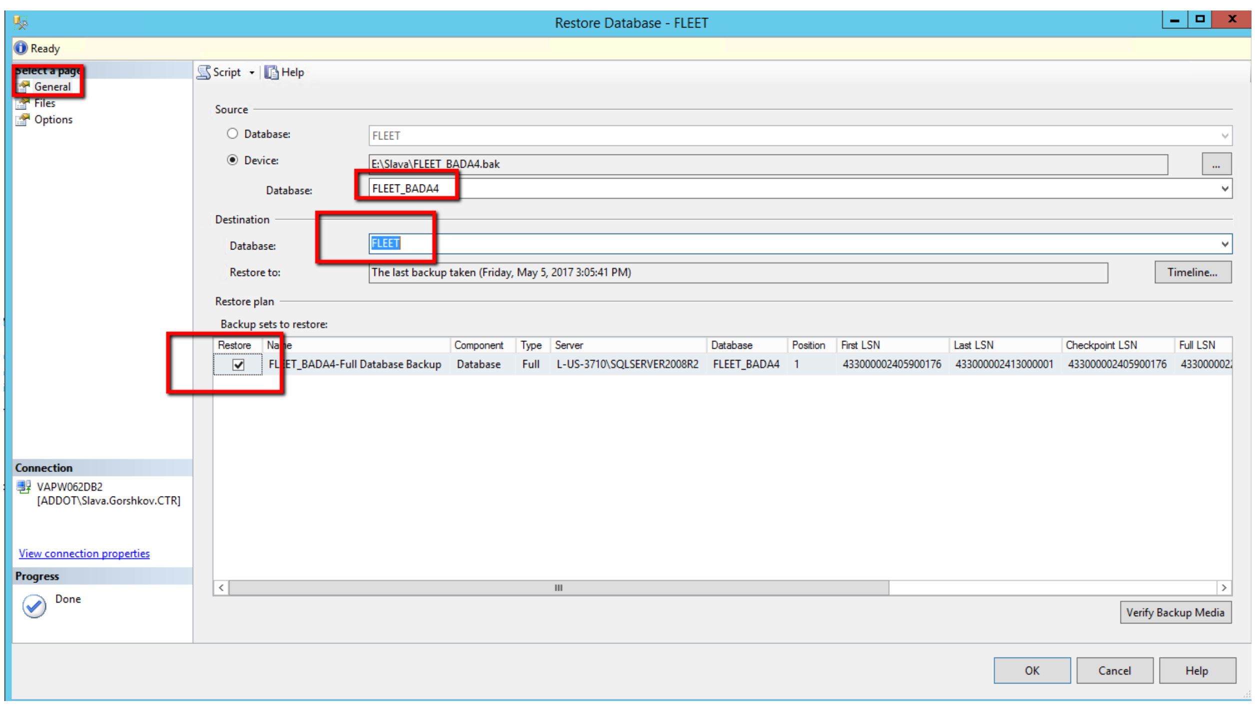
After the Restore is complete, the user should issue the following commands:

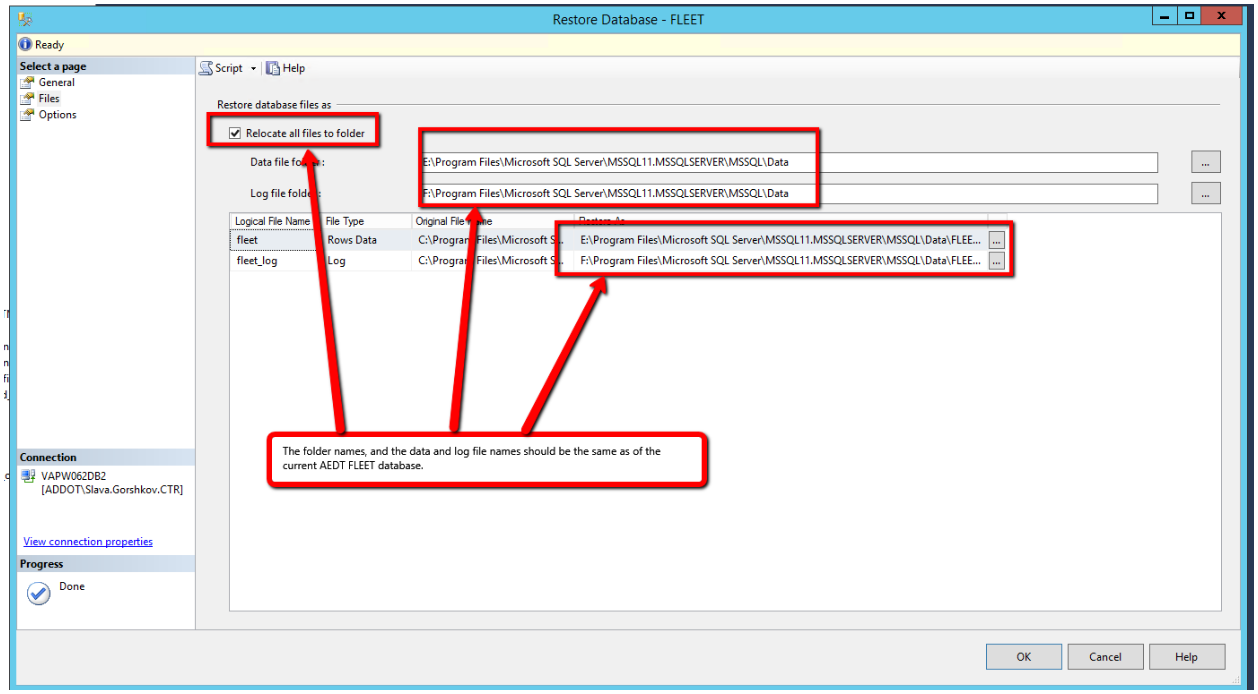
```
use master;

alter database FLEET
      set online with rollback immediate
```

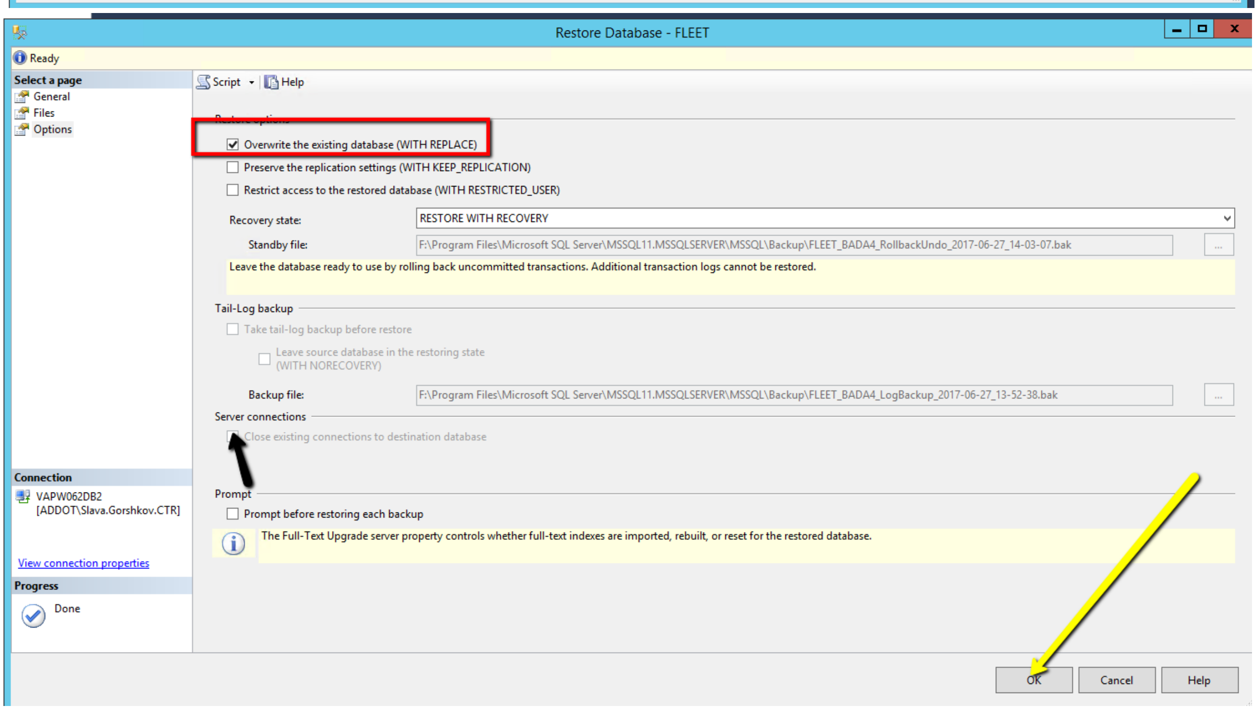
4.5. If the Restore fails, the user might try to Delete the FLEET database first, and then repeat the item # 4.3.

4.6. If the Delete fails, then the user need to restart the SQL Server services, and then repeat the item # 4.3.

5. **If the user has SSMS 2012,** the main Restore dialog looks a little different, and the process of replacing the existing FLEET database with the FLEET_BADA4 database is more reliable, because the user can explicitly choose to close all the connections to the destination database, see the last screen shot.




Restore Database - FLEET
Ready
Select a page
General
Files
Options
Script
Help
Restore database files as
Relocate all files to folder
Data file folder:
E:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data
Log file folder:
F:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data
Logical File Name
File Type
Original File Name
Restore As
fleet
Rows Data
E:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data\FLEE...
fleet_log
Log
F:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data\FLEE...
The folder names, and the data and log file names should be the same as of the current AEDT FLEET database.
Connection
VAPW062DB2
[ADDOT\Slava.Gorshkov.CTR]
View connection properties
Progress
Done
OK
Cancel
Help


Restore Database - FLEET
Ready
Select a page
General
Files
Options
Script
Help
Overwrite the existing database (WITH REPLACE)
Preserve the replication settings (WITH KEEP_REPLICATION)
Restrict access to the restored database (WITH RESTRICTED_USER)
Recovery state:
RESTORE WITH RECOVERY
Standby file:
F:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\FLEET_BADA4_RollbackUndo_2017-06-27_14-03-07.bak
Leave the database ready to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.
Tail-Log backup
Take tail-log backup before restore
Leave source database in the restoring state (WITH NORECOVERY)
Backup file:
F:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\FLEET_BADA4_LogBackup_2017-06-27_13-52-38.bak
Server connections
Close existing connections to destination database
Prompt
Prompt before restoring each backup
The Full-Text Upgrade server property controls whether full-text indexes are imported, rebuilt, or reset for the restored database.
Connection
VAPW062DB2
[ADDOT\Slava.Gorshkov.CTR]
View connection properties
Progress
Done
OK
Cancel
Help

Appendix C: Sensor Path Import Tool

Introduction

This appendix describes the Sensor Path Import (SPI) tool, which is a command-line tool that reads aircraft flight trajectory data and optionally adjusts speed, altitude, latitude and longitude using smoothing and filtering algorithms in order to reduce flight modeling failures. In the smoothing process, data noise and spurious points are removed from the trajectory. Following trajectory smoothing, these data are filtered by reducing the number of points (i.e. subsampled) while retaining characteristics and so-called change points of the original trajectory. The user has the option of smoothing input ground speeds or recalculating the speeds based on smoothed longitude/latitude coordinates. Filtering for point reduction is an optional process, and the SPI tool can be run in pass-through mode where no smoothing or filtering is applied. The processed trajectory data is then imported into the SENSOR_PATH table in an AEDT study database.

The SPI tool assumes AEDT is fully installed and operational. SPI is provided with the installation of AEDT and is located in the following directory *C:\Program Files\FAA\AEDT\SensorPathImport\SPI.exe.* The procedure for using the SPI tool is as follows:

1. Install the R engine and required packages (first use only)
2. Format trajectory data and prepare input files
3. Create an AEDT study into which the data will be imported, and add all airports used to the study.
4. Edit the SPI configuration file to specify:
    a. Study database
    b. Fleet database
    c. SQL server information
    d. Optionally, change the smoothing and filtering settings from default values
5. Run the SPI tool from a command prompt using desired options.

The steps are described in greater detail below. At present, there is no way to run the smoothing and filtering process separately from the database import process. If the only goal is to smooth/filter trajectory data, it is possible to create a new AEDT study to simply receive the imported data and use the –c flag (described below) to save the output to a file.

Install R Engine

1. Download R package for Windows from here: https://cran.cnr.berkeley.edu/
2. Install with all defaults (including both 32 and 64 bit installations)
3. Start “R x64” version of the program as administrator.

4. In the command window that appears, execute the following commands:

```
install.packages('pspline')
install.packages('sp')
install.packages('geosphere')
```

Format trajectory data and prepare input files

Prepare input files

The SPI tool requires two input files: a trajectory file and an operations file. These files must be in CSV format, and should not contain empty rows.

Trajetory File Format

The input trajectory file contains trajectory data for one or multiple flights in CSV format. It contains individual point data, one line at a time, that are grouped by flight identification number (FID). Within a given FID grouping, the point data must be sorted by **Time** in ascending order; duplicate Time entries will be skipped; otherwise, the SPI tool will exit with error message unless user specifies **–n** (no smoothing) as a command-line argument. The trajectory file columns are described in the table below:

Column Name	Column Description
FID	Flight ID. Must be unique for individual flights. If duplicate flight ids (FID) exist in the file, the SPI tool will exit with error message.
SeqId	Sequence Id
Long	Longitude
Lat	Latitude
GpsAlt	GPS Altitude (Feet) Note: Only the GPS Altitude field can be used in AEDT performance modeling. Because users will typically only have pressure altitude data, users may want to use pressure altitude data as a surrogate for GPS altitude.
Time	Time. Must be in the following format: YYYY-MM-DDThh:mm:ss.fff (UTC)
One of the following speed types: • GS – Ground speed (Knots); • KTAS – True air speed (Knots); • KCAS – Calibrated air speed (Knots); or • MACH – MACH number (unitless)	Only one speed type can be present in the file.
AltP	Pressure Altitude (Feet)

Column Name	Column Description
	Note: Only the GPS Altitude field can be used in AEDT performance modeling. Because users will typically only have pressure altitude data, users may want to use pressure altitude data as a surrogate for GPS altitude.

Notes

- The SPI tool is able to process a single trajectory file with data on 30M flights.
- The date part of a time stamp is required for MESSAGE_TIME column in the SENSOR_PATH table and also for creating a unique flight id (by combining FID with integer part of Longitude, Latitude, and full value of a Time column (e.g. "S562_-105_46_2017-04-03T06:33:28.137")).

Trajectory File Example

```
FID,SeqId,Long,Lat,GpsAlt,Time,GS,AltP
383662,1,-73.7726517,40.6283989,26.1,2017-10-27T03:53:17.999 , 9.4,-212.0
383662,2,-73.7726974,40.6284180,26.4,2017-10-27T03:53:19.000 ,12.1,-212.0
383662,3,-73.7727737,40.6284485,26.2,2017-10-27T03:53:20.001 ,15.1,-210.0
............................
.......TRUNCATED............
............................
383662,1023,54.6660309,24.4225178,105.1,2017-10-27T15:49:43.000 ,139.6,34.0
383662,1024,54.6654854,24.4229240,99.9,2017-10-27T15:49:44.000 ,139.1,18.0
383662,1025,54.6649399,24.4233303,89.6,2017-10-27T15:49:45.004 ,138.4,14.0
383810,1,-73.7654495,40.6477928,34.2,2017-10-28T03:30:24.998 , 9.4,34.0
383810,2,-73.7654724,40.6477547,34.2,2017-10-28T03:30:25.999 ,11.1,33.0
383810,3,-73.7655182,40.6477013,34.2,2017-10-28T03:30:27.000 ,14.2,41.0
```

Operation File Format

The input operation file contains operations data for each flight in the trajectory file, i.e. one line in the operation file per each flight. The input columns are described below.

Column Name	Column Description
FID	Flight id from the input trajectory file.
OpID	User-defined Operation ID.
AircraftType	ICAO aircraft type from FLT_ACTYPES table. Used to map it to Equipment ID. User can override default mapping with an explicit value in the optional **EquipId** column.
EquipId	[Optional] Equipment ID from FLT_EQUIPMENT table.
NumOps	Number of times to fly this operation.
DepApt	Departure airport ICAO code from APT_CODE table. The airport must be already in the study and have at least one airport layout and at least one runway.
DepRwy	Departure runway end name from APT_RWY_END table

Column Name	Column Description
ArrApt	Arrival airport ICAO code from APT_CODE table. The airport must be already in the study and have at least one airport layout and at least one runway.
ArrRwy	Arrival runway end name from APT_RWY_END table.
OffTime	Wheels-off time. Required for full flight and partial departure operation types. Empty for partial arrivals.
OnTime	Wheels-on time. Required for full flight and partial arrival operation types. Empty for partial departures.
Profile	[Optional] Aircraft's profile ID from FLT_ANP_AIRPLANE_PROFILES table.
StageLen	[Optional] Stage length.
TaxiIn	[Optional] Taxi-in time (in minutes – will be converted and stored in seconds).
TaxiOut	[Optional] Taxi-out time (in minutes – will be converted and stored in seconds).
CruiseAlt	[Optional] Cruise altitude of the aircraft (Feet).
DepGate	[Optional] Departure gate name from APTLAYOUT_GATE table.
ArrGate	[Optional] Arrival gate name from APTLAYOUT_GATE table.
Operator	[Optional] Operator – future use.
RegNum	[Optional] Registration number – future use.
FlightNum	[Optional] Flight number – future use.
BeaconCode	[Optional] Beacon Code (octal number, treated as string) – future use.
ModeSCode	[Optional] Mode-S Code - future use

Notes

- OffTime and OnTime fields must be in the following format: YYYY-MM-DDThh:mm:ss
- If both **OffTime** and **OnTime** are present for the operation, the Operation type is set to Full Flight - Runway-to-Runway.
- If only **OffTime** is present, the Operation type is set to Partial Departure.
- If only **OnTime** is present, the Operation Type is set to Partial Arrival.

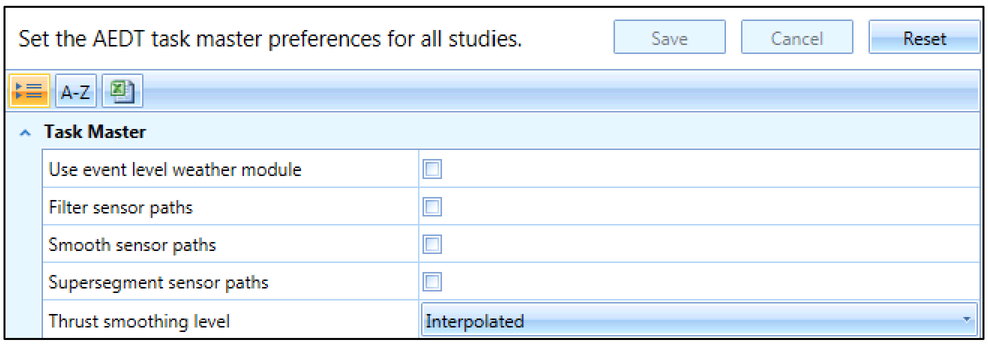
Operation File Example

```
FID,OpID,AircraftType,EquipId,NumOps,DepApt,DepRwy,ArrApt,ArrRwy,OffTime,OnTime,Profile,StageLen,
TaxiIn,TaxiOut,CruiseAlt,DepGate,ArrGate,Operator,RegNum,FlightNum,BeaconCode,ModeSCode
383662,383662,B777,,1,KJFK,31L,OMAA,31L,2017-10-27T03:53:17,2017-10-27T15:49:45,,,,,,,,,,,
383810,383810,B777,,1,KJFK,22R,OMAA,13R,2017-10-28T03:30:25,2017-10-28T15:21:36,,,,,,,,,,,
384001,384001,B777,,1,KJFK,31L,OMAA,31L,2017-10-29T03:31:32,2017-10-29T15:02:50,,,,,,,,,,,
385136,385136,B777,,1,KJFK,31L,OMAA,31L,2017-10-30T03:45:43,2017-10-30T15:43:10,,,,,,,,,,,
385698,385698,B777,,1,KJFK,22R,OMAA,31L,2017-10-31T03:48:36,2017-10-31T16:00:09,,,,,,,,,,,
```

Configure the AEDT Study

The SPI tool requires a study database into which data will be imported. Before running the tool, use the AEDT user interface to create a study and add airports. Every airport specified in the operations file must be added to the study before running the SPI tool.

Additional filtering in AEDT may be desired beyond what the SPI tool provides. The *Filter sensor paths* option (described in the AEDT Technical Manual in Section 3.7.2.1) is available in the AEDT *Study* tab, *Preferences*, *Task Master* screen.



Edit the SPI Configuration File

1. Change the TargetStudy and Fleet Database Connection Strings Open the *SPI.exe.config* file located in *C:\Program Files\FAA\AEDT\SensorPathImport*. Administrative privileges will be required to save changes to this file.
2. Change the **TargetStudy** connection string. The imported trajectories and operations will be stored in the AEDT study specified after “Initial Catalog=”.
3. Change the **Fleet** connection string. The Fleet database is used by the SPI tool to map aircraft types in the operations file.

```
<connectionStrings>
    <add name="TargetStudy" connectionString="Data Source=(local);Initial
Catalog=MyStudy;Integrated Security=True" providerName="System.Data.SqlClient" />
    <add name="FLEET" connectionString="Data Source=(local);Initial
Catalog=FLEET;Integrated Security=True" providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Smoothing and filtering config file settings

Smoothing and filtering settings can be changed from their defaults by editing values in the

SPI.exe.config file. A full list of the settings is described below.

Setting	Default	Description
altWindow	11	Altitude smoothing: Hampel filter window width. Expected integer > 0 Value is half width: 11 points corresponds to 11+11+1=23 total points
altOutlier	3	Altitude smoothing: outlier criterion. Expected float > 0 Units are in terms of standard deviation σ within Hampel filter window
altTol	200	Altitude filtering: RDP tolerance. Expected float > 0 Tolerance in feet
spdWindow	11	Speed smoothing: Hampel filter window width. Expected integer > 0 Value is half width: 11 points corresponds to 11+11+1=23 total points
spdOutlier	3	Speed smoothing: outlier criterion. Expected float > 0 Units are in terms of standard deviation σ within Hampel filter window
spdAveWindow	5	Speed smoothing: averaging window width / point reduction ratio Expected integer > 0. Ratio of N:1 retains every Nth point
distTol	0.10	Ground track filtering: RDP tolerance. Expected float >= 0 Tolerance in nmi
filtFlag	true	Boolean switch for down-sampling. Expected Boolean (true/false) true activates filtering algorithms to reduce number of points

Running the Sensor Path Import (SPI) Tool

To invoke the SPI tool, open a command prompt window, navigate to the *SensorPathImport* folder, and type **SPI.exe** (or just SPI) with the argument list. As with AEDT, note that the user must be an administrator to successfully run the SPI tool. If no arguments are given or if they cannot be parsed, the tool will print the usage message and exit. The SPI tool must be run from the *SensorPathImport* folder so that dependent files can be located.

Command line options

The list of valid switches with arguments:

`Switches and arguments`	Description
`-t` `trajectoryFilename`	Use this switch to specify a trajectory file.
`-o` `operationFilename`	Use this switch to specify an operation file.
`-d` `DatabaseServer:DatabaseName`	Use this switch to specify DB server & DB name where the **MVT_RAW_ETMS_POSITIONS** table is located. Population of the table is done by the SPI Tool.

Switches and arguments	Description
-n	Use this switch for no smoothing (aka "pass-through mode"). The option is provided mainly for: (1) debugging, (2) getting high-fidelity data (e.g., CFDR) through the AEDT-BADA4 sensor path workflow, and (3) the AEDT-BADA3 sensor path workflow. In these cases, smoothing is either not required or not desired.
-r	Use this switch if you want the smoothing algorithm for speed to recalculate speed after lat/long and altitude smoothing. **–r** switch has no effect if **–n** (no smoothing) switch is also used.
-c	Use this switch to save output of the processing to a CSV file. The file will be saved in the same folder where SPI.exe is located. The option is provided mainly for debugging and currently has a limit of 1,048,576 trajectory points. For every 2,000 flights, their trajectories will be saved in the separate file with chunk number appended. SPI tool reads flight trajectories and processes them in chunks of 2,000 (because of memory and R environment limitations).

There are two main scenarios where the SPI tool can be used.

1. Read sensor path data from a CSV trajectory file and create records in the SENSOR_PATH table as well as matching records in the AIR_OPERATION table using information from the CSV operation file. For this scenario user has to provide at least two switches: –t and –o.
   ```
   SPI –t c:\traj.csv –o c:\oper.csv
   ```

2. Read sensor path data from the MVT_RAW_ETMS_POSITIONS table in a database. In this case, it only creates records in the SENSOR_PATH table, and not in the AIR_OPERATION as there is no matching operation data. Database server and database name are listed after the **–d** switch and must be separated by a colon:
   ```
   SPI –d VAPW062DB2:MCM
   ```

–d and **–t/–o** switches are mutually exclusive; if provided together the tool will exit with usage message.

If trajectories are read without errors, the SPI tool will start smoothing and filtering them unless **–n** (no smoothing) switch is given on command line.

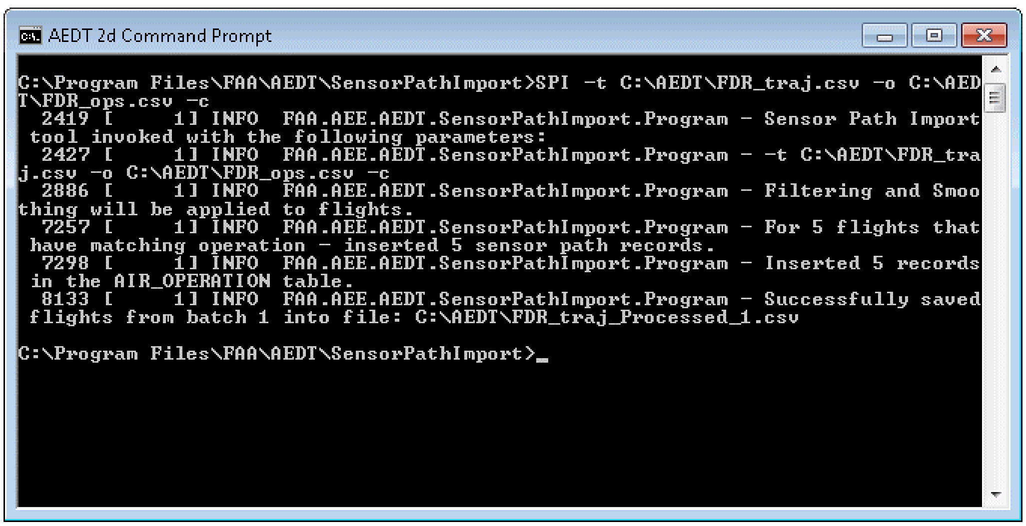
Examples

Example 1: basic import with smoothing / filtering and file output

`SPI -t c:\traj.csv -o c:\oper.csv -c`
will save results to *c:\traj_Processed_1.csv* (for smoothed trajectories from the first chunk).

Example 2: basic import with file output and without smoothing / filtering

`SPI -t c:\traj.csv -o c:\oper.csv -c -n`
will save results to *c:\traj_Raw_1.csv* (unmodified trajectories from the first chunk)



Example 3:

`SPI -d SQLServerName:DBname`
will save results to:

- SQLServerName_DBname _Processed_1.csv (for smoothed trajectories from the first chunk); ***or***
- SQLServerName_DBname _Raw_1.csv (if **–n** switch is applied)
-

Notes

During runtime, the console will automatically output the definition of the Smoothing/Filtering Function, but the user can ignore this output (as it does not necessarily reflect the actual settings that the user provided). Instead, see the log file.

If successfully processed, the resulting trajectories are then stored in the study specified in the **connectionStrings** in SPI.exe.config.

```
<connectionStrings>
    <add name="TargetStudy" connectionString="Data Source=localhost;Initial
Catalog=STUDY_INM;Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
```

Troubleshooting common SPI errors

1. Problem: the SPI tool cannot access the study database. If the study was not created before running the SPI tool, or if an incorrect study name is specified in *SPI.exe.config*, an INFO message followed by an ERROR message describing a login failure will be displayed in the command window:

   ```
   219 [      1] INFO  FAA.AEE.AEDT.DataAccessModule.Cache.AirportCacheSingleton -
   Missing STUDY_B763 in server database. SQL Server: Data
   Source=localhost;Initial Catalog=STUDY_B763;Integrated Security=True
   14949 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.OperationFileParser - Cannot
   open database "STUDY_B763" requested by the login. The login failed.
   Login failed for user 'AEDT_user'.
   ```

   Solution: ensure that the study exists and can be opened in AEDT, and check that the study name (STUDY_B763 in this example) is specified as follows in the *SPI.exe.config* file:

   ```
   <connectionStrings>
          <add name="TargetStudy" connectionString="Data Source=localhost;Initial
   Catalog=STUDY_B763;Integrated Security=True"
   providerName="System.Data.SqlClient" />
   </connectionStrings>
   ```

2. Problem: operations or trajectory files cannot be accessed by the SPI tool. If the operations or trajectory files are open in another program, the following error message will be returned:

   ```
   3399 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.OperationFileParser - Could
   not parse operation file: C:\SPI\FDR_operation_Test.csv
   3409 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.OperationFileParser - The
   process cannot access the file 'C:\SPI\FDR_operation_Test.csv' because it is
   being used by another process.
   ```

   Solution: make sure that both operations file and trajectory file are closed and re-run the SPI tool.

3. Problem: incorrect or incomplete SQL server name given in *SPI.exe.config* file.

   ```
   33779 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.OperationFileParser - A
   network-related or instance-specific error occurred while establishing a
   connection to SQL Server. The server was not found or was not accessible.
   ```

```
Verify that the instance name is correct and that SQL Server is configured to
allow remote connections. (provider: SQL Network Interfaces, error: 26 - Error
Locating Server/Instance Specified)
```

Solution: The SQL server instance must be specified after "Data source=" in the connection string entries in *SPI.exe.config*. This will depend on how SQL server is configured; typical entries are of the following forms:

```
Data Source=localhost
Data Source=localhost\SQLEXPRESS12
Data Source=[Computer name]\SQL2012
```

4. Problem: input error in Windows command line. If the pathnames have special characters or spaces which are not properly entered, the SPI tool may throw the following error:

   ```
   2859 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.Program - Input trajectory
   file does not exist.
   ```

   Solution: enter the pathnames correctly in the Windows command prompt. In this example, the directory containing the trajectory file includes a space, and the error is generated with the following command:

   ```
   spi -t C:\SPI test\FDR_Data_SPI_Format.csv -o C:\SPI
   test\FDR_operation_Test.csv
   ```

   Enclosing the pathnames with quotation marks allows the SPI tool to load the files as intended:

   ```
   spi -t "C:\SPI test\FDR_Data_SPI_Format.csv" -o "C:\SPI
   test\FDR_operation_Test.csv"
   ```

5. Problem: Incorrect AircraftType specified in operations file. If an invalid aircraft type is specified in the operations file and an equipment ID is not used, the following error message will be shown:

   ```
   4849 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.OperationFileParser -
   Skipping operation. Cannot get aircraft ID for UFID: 1801
   4869 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.Program - Skipping processing
   1801 – no matching operation found.
   ```

   Solution: Ensure that the AircraftType and/or EQUIP_ID match the intended aircraft.

6. Problem: missing airport layout. If all airports used in flights tabulated in the operations file have not been added to the study within AEDT, the SPI tool will throw an error message:

   ```
   4879 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.DataImporter - Could not get
   airport layout ID from KLAX
   4879 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.DataImporter - Value cannot
   be null.
   Parameter name: source
   4879 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.OperationFileParser -
   Skipping operation. Cannot get layout ID and location from DepApt/DepRwy
   values: KLAX/07L
   ```

   Solution: add airport layouts to the study under the "Airports" ribbon group in AEDT, and then re-run the SPI tool.

7. Problem: empty rows exist in either the operations file or trajectory file. If the operations files contains an empty row, the SPI tool will stop processing operations after encountering an empty row and thus, not all of the trajectory data will be inserted in the SENSOR_PATH table. If the

trajectory file has an empty row, the SPI tool will likewise stop processing trajectory data and the following error message may be given:

```
4939 [      1] ERROR FAA.AEE.AEDT.SensorPathImport.Program - Index was outside
the bounds of the array.
```

Solution: check that both the trajectory file and operation file do not contain empty rows.

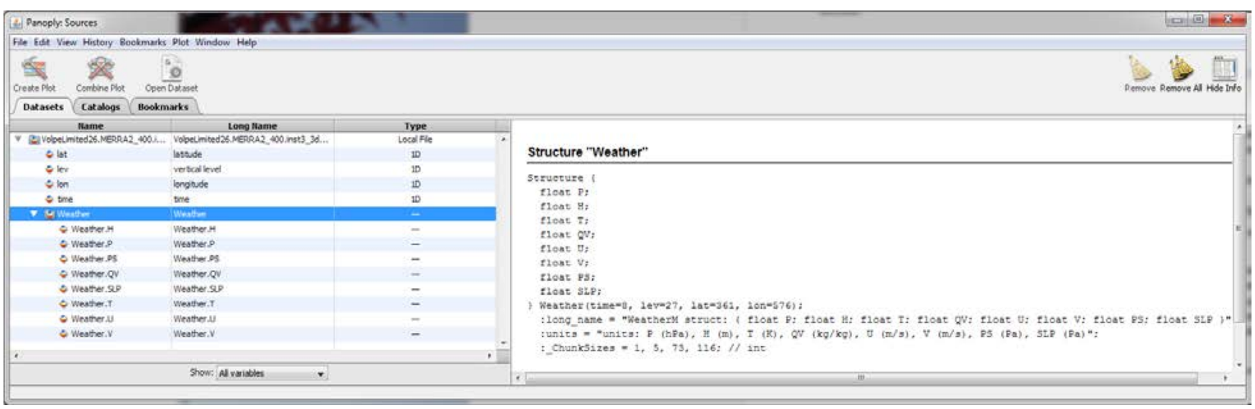
Appendix D: Acquiring and Preparing MERRA-2 Weather Data for AEDT Modeling

Introduction

From MERRA2 weather data, we need to retrieve seven variables – T (temperature), H (geopotential height), QV (specific humidity), U (eastward-wind), V (northward-wind), PS (surface pressure), SLP (sealevel pressure) – from 3D data. This means that we would have to call seven NetCDF4 API calls to retrieve the weather data at a particular grid lat/lon point (pressure levels are obtained automatically) if AEDT used the native NetCDF4 variable based approach. For AEDT, we reorganize/transform the MERRA2 weather data into a non-variable but a chunked (struct) variable based NetCDF4 data so that we call just one API (instead of seven APIs) to retrieve 42 (or less) levels of weather data.

The **NC4WXEditorWPF.exe** is the application which transforms the MERRA2 weather files into the chunked format usable by AEDT.

The caveat is that once you reorganize the data, the generic NetCDF file viewers cannot show the content of the chunked data. For example, the Java based Panoply tool can show how the variable is defined with attributes but cannot show each content inside the struct.

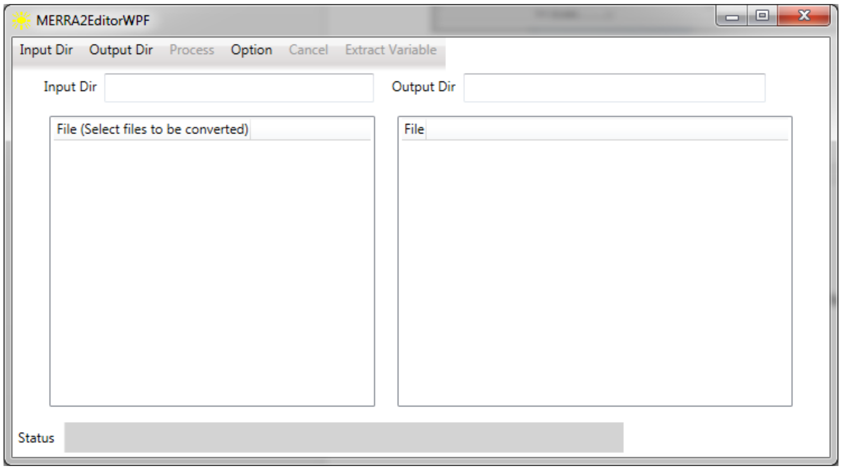


How to use NC4WXEditorWPF.exe

Note that before using the NC4WXEditor, you must have Visual C++ 2015 Redistributable installed. If not, you can download this at www.microsoft.com/en-us/download/details.aspx?id=53587. If

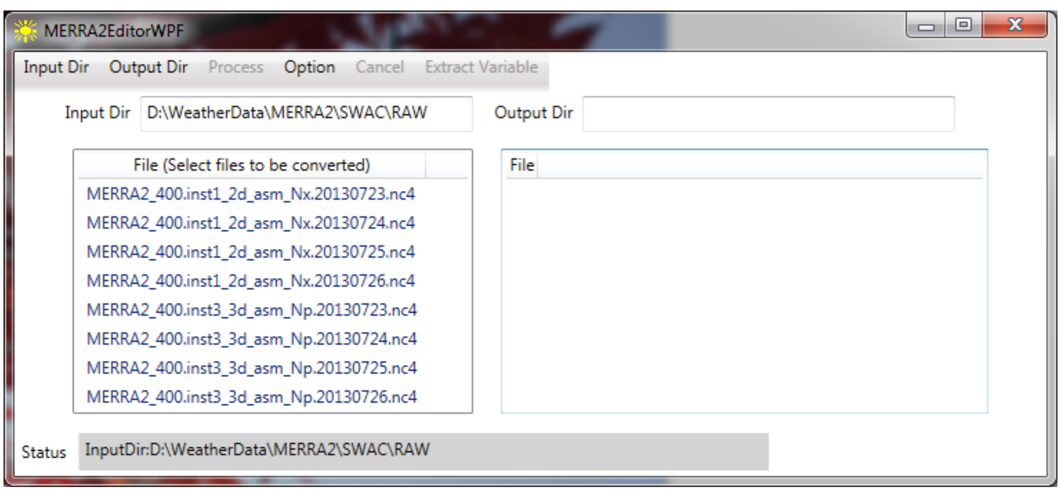
NC4WXEditorWPF still does not open after this installation, check that FAA.AEE.AEDT.MERRA2CLI.dll, Microsoft.Expression.Interactions.dll, and System.Windows.Interactivity.dll are all installed on your computer.

1. Start the application. The executable is located in the NC4WXEditorWPF subfolder in the AEDT installation directory.

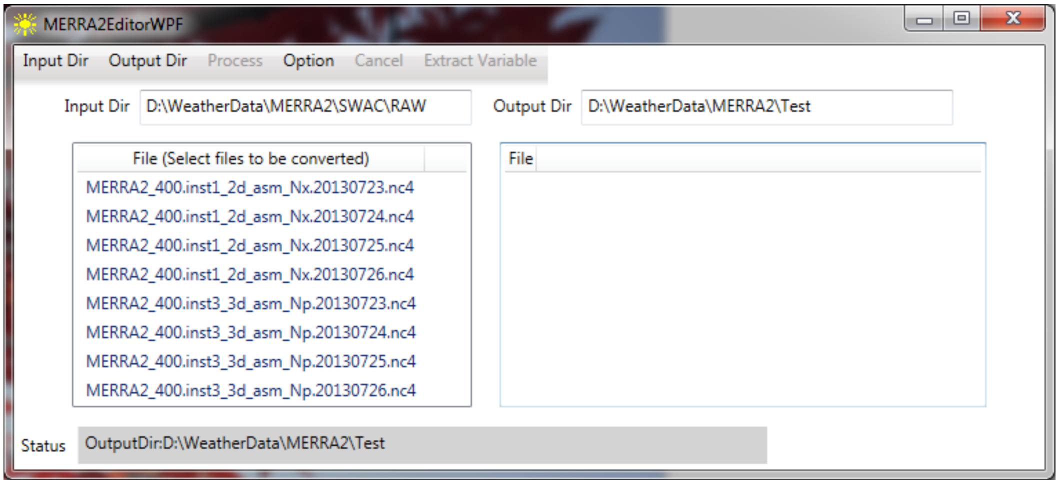


2. Select the input directory where the relevant MERRA2 files are present (may contain both 2d and 3d files) by clicking the **Input Dir** menu. This will bring up the **Browse For Folder** dialog.

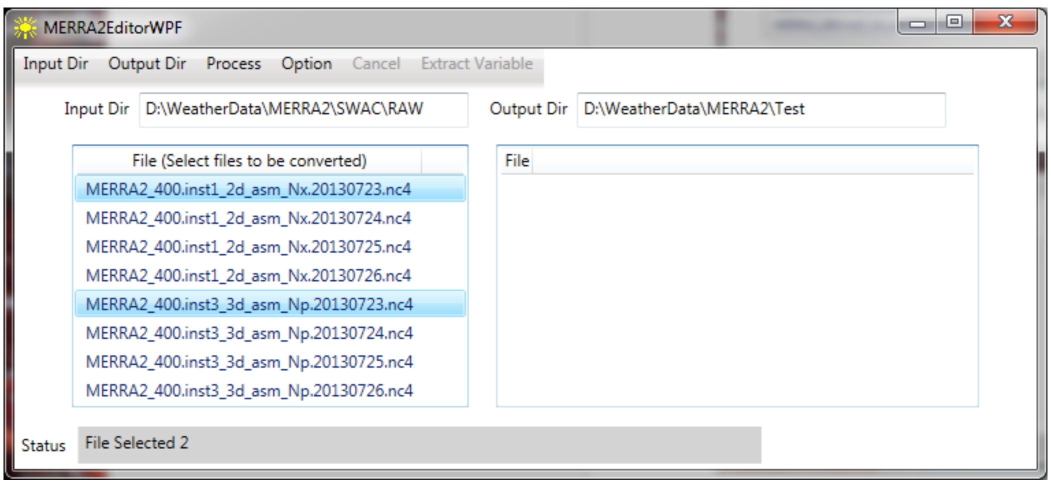
3. After selecting a folder, identified files (only those usable in AEDT) are listed as follows:



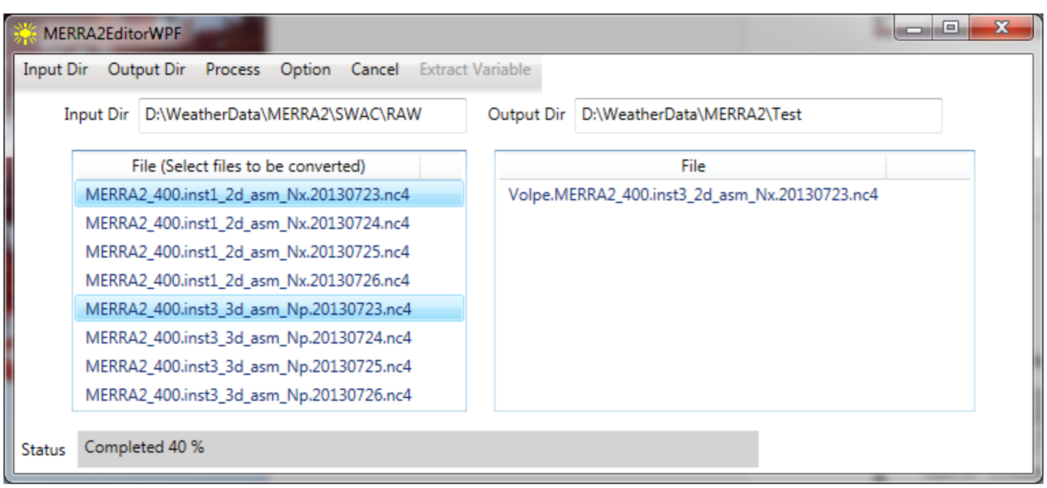
4. Click the **Output Dir** menu. In the **Browse For Folder** dialog, select a folder. The selected folder is listed in the Output Dir text box.



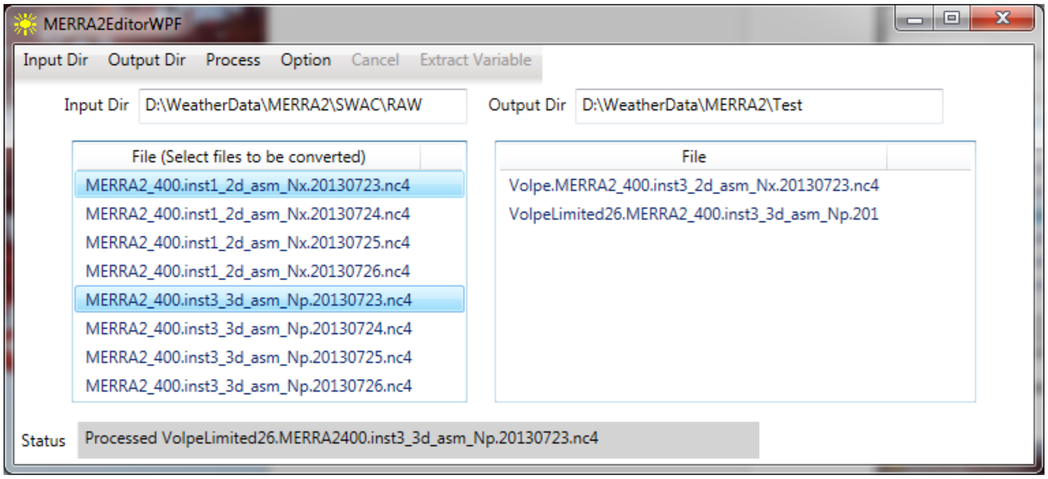
5. Pick the files to be converted. Two files (2d and 3d) for 7/23/2013 are selected below.



6. Click the **Process** menu. The 2d file processing is very fast. The 3d file processing takes time. The status bar at the bottom shows the percentage complete for each file.

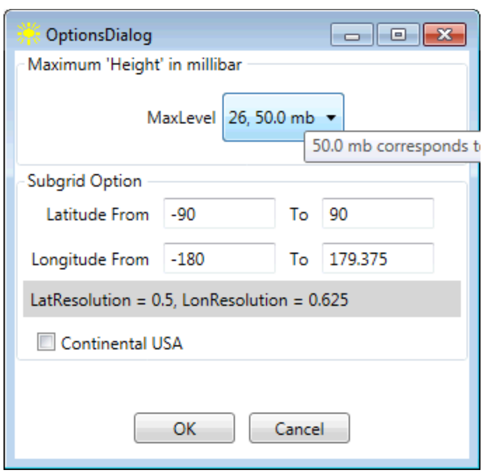


7. When the processing is complete, the following screen is displayed. The MERRA2 files have been transformed into chunked NetCDF data (reformatted so that they are usable in AEDT).



Options for reducing the file size

Click the **Option** menu to open the Options dialog.



Reduce the file size by limiting the number of layers

The **MaxLevel** option allows changing the number of layers from 42 to a smaller number. The "**26, 50.0 mb**" is selected by default – this will ignore 50 mb or smaller pressure level layers.

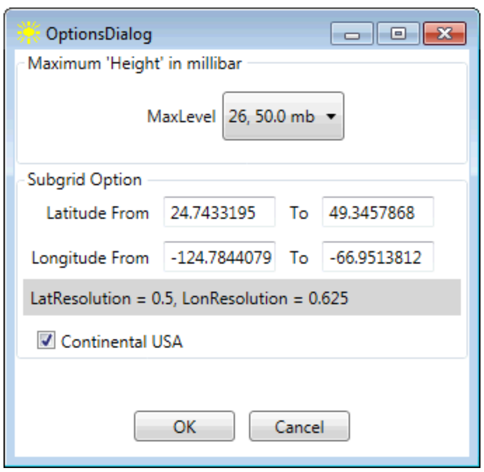
The selected MaxLevel will be reflected in the processed 3d file name – for example:

- VolpeLimited26.MERRA2_300.inst3_3d_asm_Np.20060101.nc4
- VolpeLimited10.MERRA2_300.inst3_3d_asm_Np.20060101.nc4

The reason for changing the MaxLevel is to make file size smaller, since the pressure levels goes up to 0.1 mb. The max height of 50 mb is around 63000 feet or 19300 meter. The original 3d data file is around 1.15 GB. After processing, the new file is around 790 MB. The file is already compressed, using NetCDF4 compression flag of 1 (fast compression).

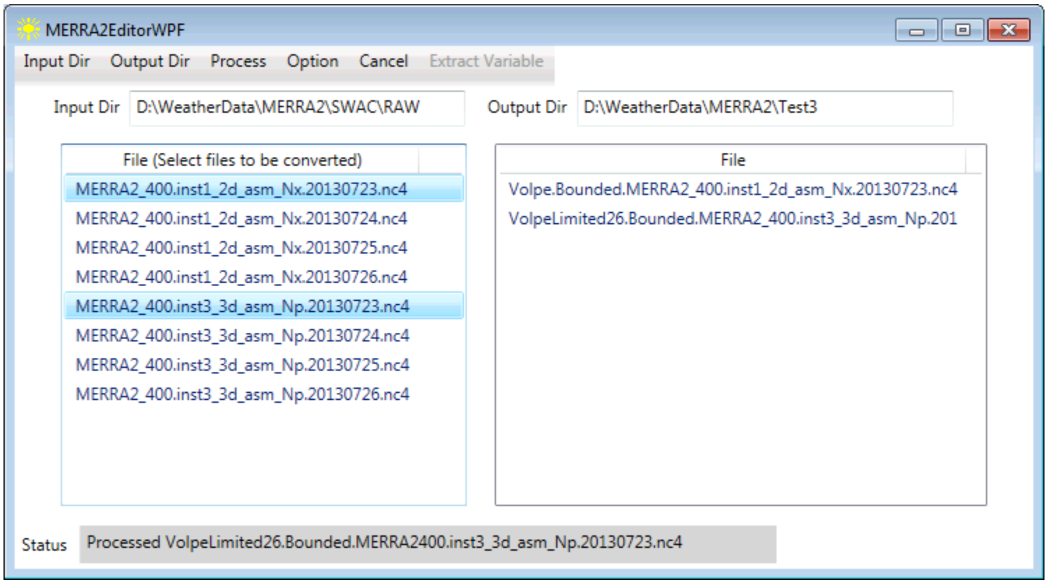
Reduce the file size by limiting the region of interest (Subgrid option)

The raw MERRA2 weather covers the entire earth: latitude from -90 to 90 degrees and longitude from -180 to 179.375. Since the longitude resolution is 0.625 degrees, the next point on the longitude is 180 degrees (i.e. -180 degrees). For convenience, the **Continental USA** checkbox can be checked to limit the region.



When the **Continental USA** option is checked, the processed files will have the “**Bounded**” in the file name:

- “Volpe.Bounded” for 2D files; and
- “VolpeLimited26.Bounded” for 3D files



The processed file Global attributes are modified to contain the reduced region:

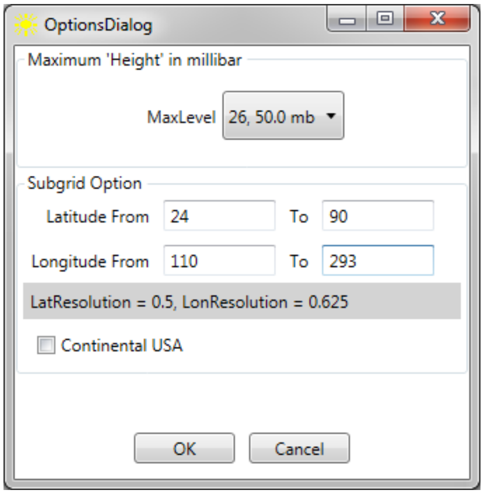
```
:Title = "MERRA2 inst3_3d_asm_Np: 3d,3-Hourly,Instantaneou
:SouthernmostLatitude = "24.500000";
:NorthernmostLatitude = "90.0";
:WesternmostLongitude = "-125.000000";
:EasternmostLongitude = "-66.875000";
:LatitudeResolution = "0.5";
:LongitudeResolution = "0.625";
:DataResolution = "0.5 x 0.625 (42 pressure levels)";
```

One interesting way to limit the region of interest is to cover the pacific region which goes over the time zone line (i.e. beyond 180 degrees). This is useful for covering the flights from Asia to USA. In order to do this in the Options dialog, the “To” value must be greater than the “From”.

For example, US East Coast is -67 degree which must be entered as 293 (= -67 + 360); so that the global attributes are written as follows:

```
:SouthernmostLatitude = "24.000000";
:NorthernmostLatitude = "90.0";
:WesternmostLongitude = "110.000000";
:EasternmostLongitude = "293.125000";
```

When you specify the upper longitude as 293, it will convert it to 293.125 due to the longitude quanta being 0.625 in MERRA2 weather data. There is no need to use the positive number for longitude for the flight location, since it automatically handles the conversion.



How to download MERRA-2 data

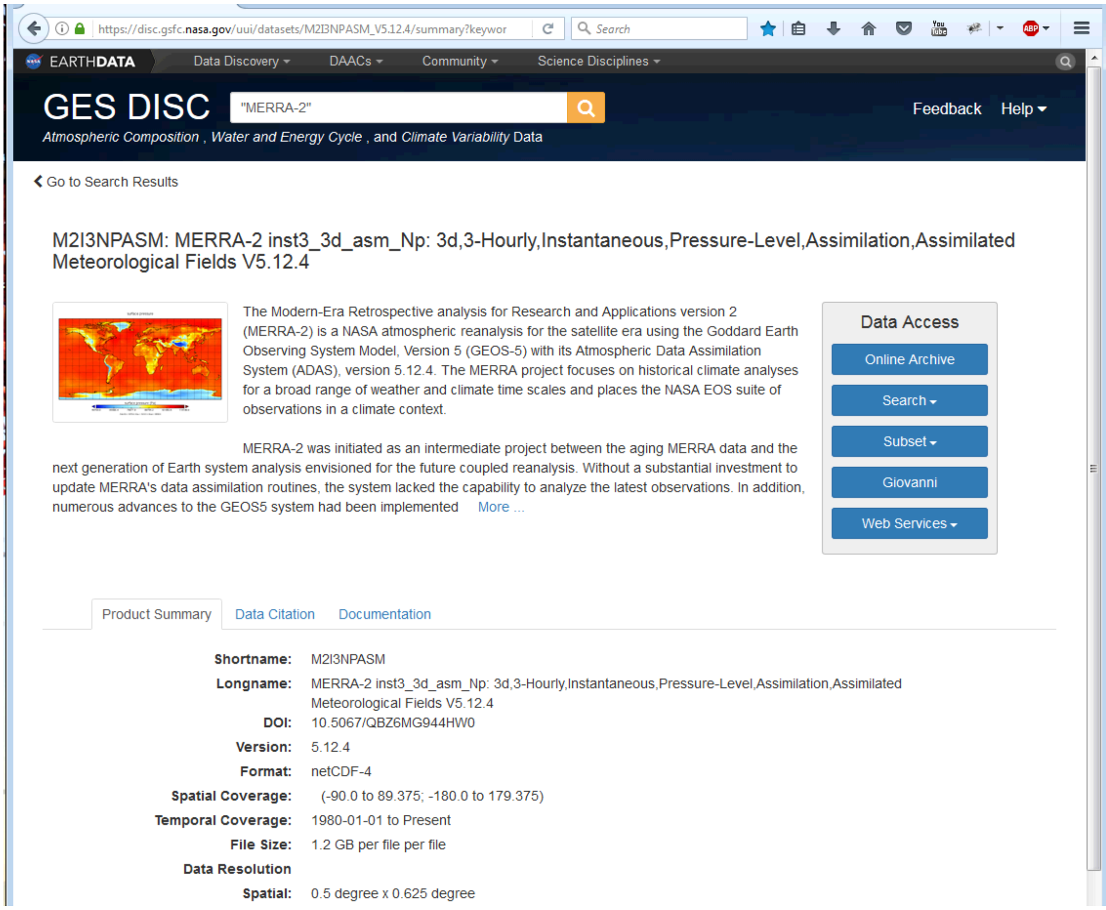
This section discusses the steps required to acquire and download the MERRA-2 weather data.

1. Create an Earthdata Username and password by registering at the following URL: https://urs.earthdata.nasa.gov
2. After registration, log in at the following address, replacing (username) in the address with the username created in step one. https://urs.earthdata.nasa.gov/users/(username)/apps
3. Follow the instruction at the link below to add the “NASA GESDISC DATA ARCHIVE” to your application:
4. https://disc.sci.gsfc.nasa.gov/registration/authorizing-gesdisc-data-access-in-earthdata_login
5. The MERRA-2 URL is: https://disc.gsfc.nasa.gov/uui/datasets?keywords=%22MERRA-2%22

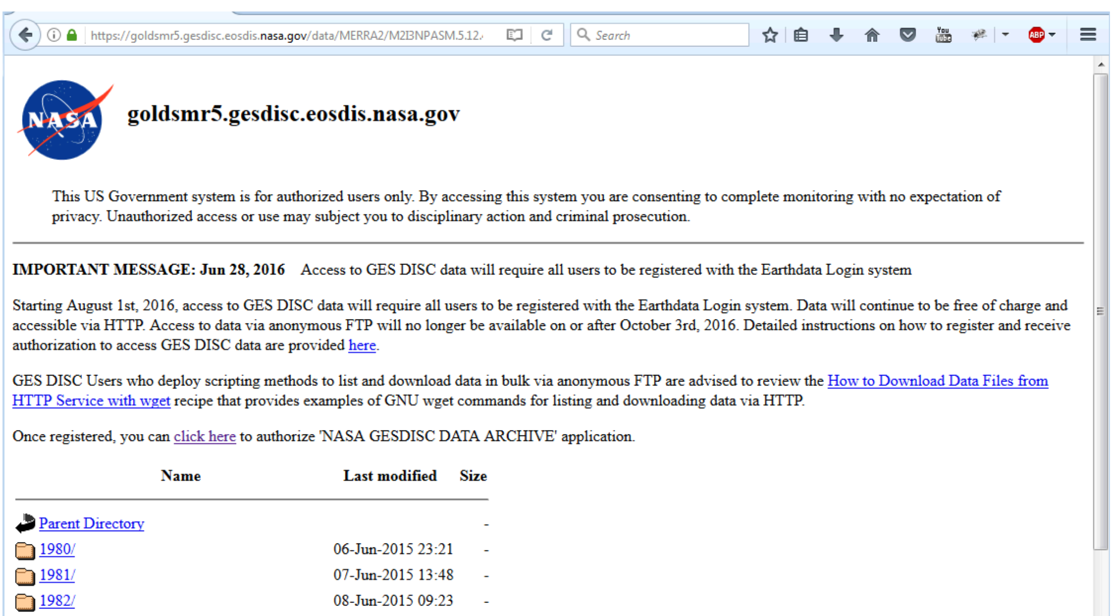
Note that AEDT uses the “inst3_3d_asm_Np” weather file for the 3D weather and the “inst1_2d_asm_Nx” weather file for the surface (2D) weather.

The 3D weather can be downloaded from:
https://disc.gsfc.nasa.gov/uui/datasets/M2I3NPASM_V5.12.4/summary?keywords=%22MERRA-2%22

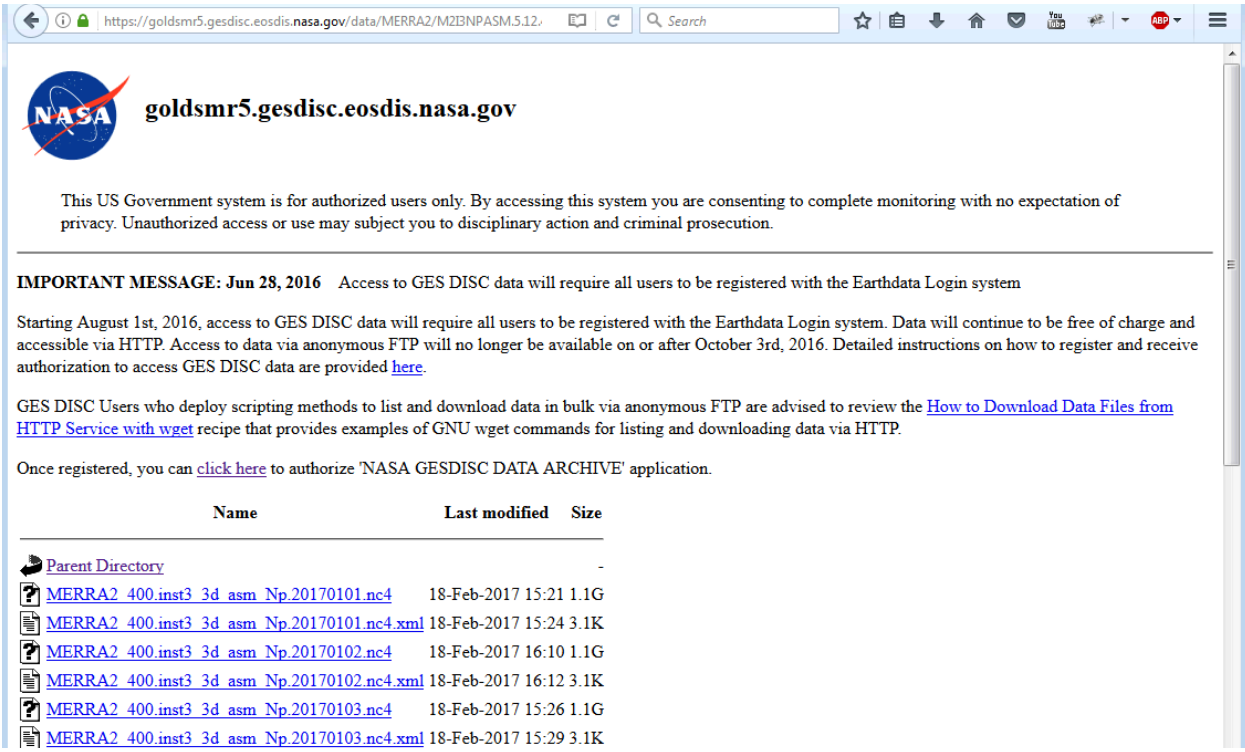
The link above should bring you to the following site:



Selecting the “Online Archive” box in the link above moves to the following site:

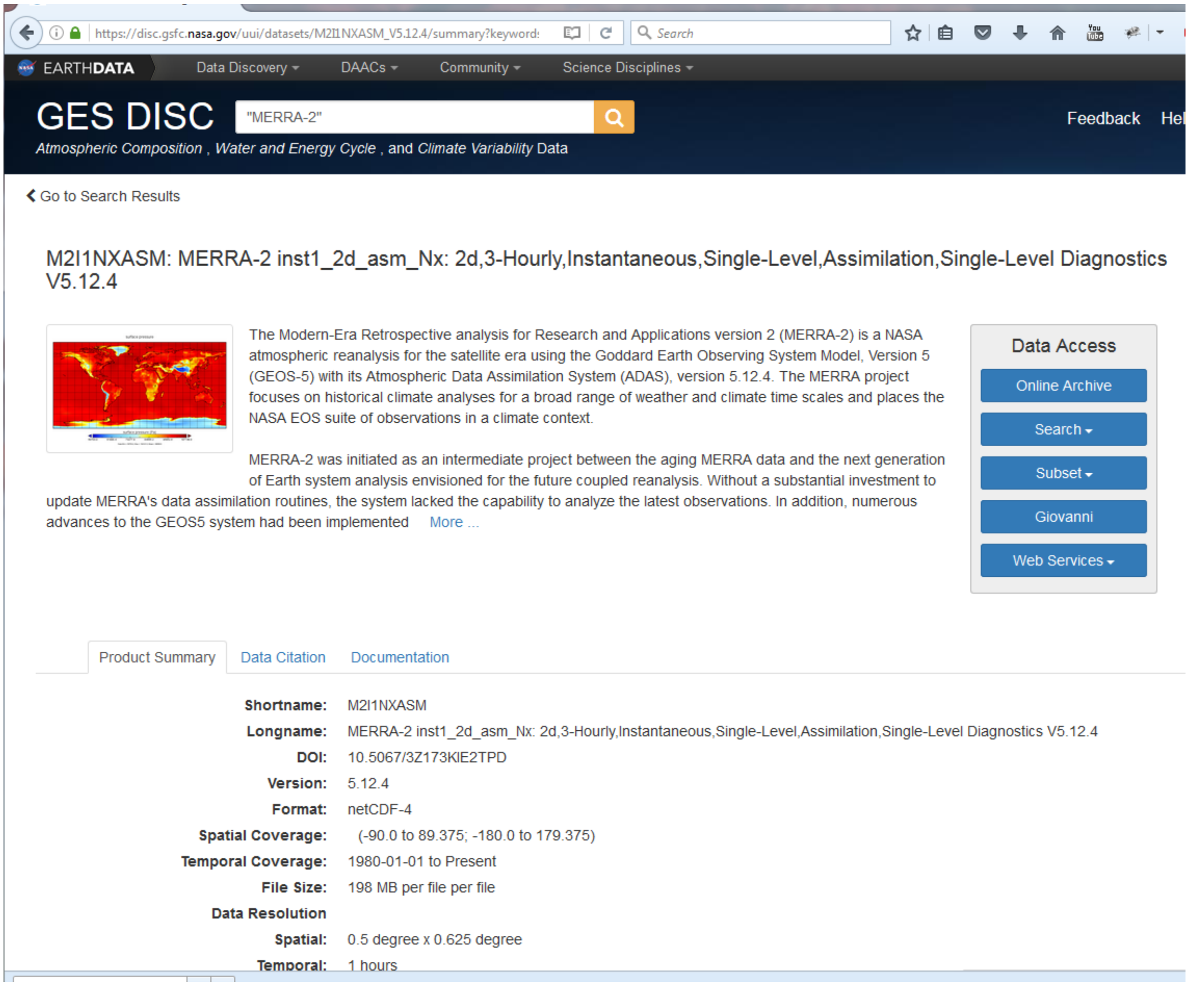


At this point, you can select the year and the month of the data you are seeking. You can then select the individual date:

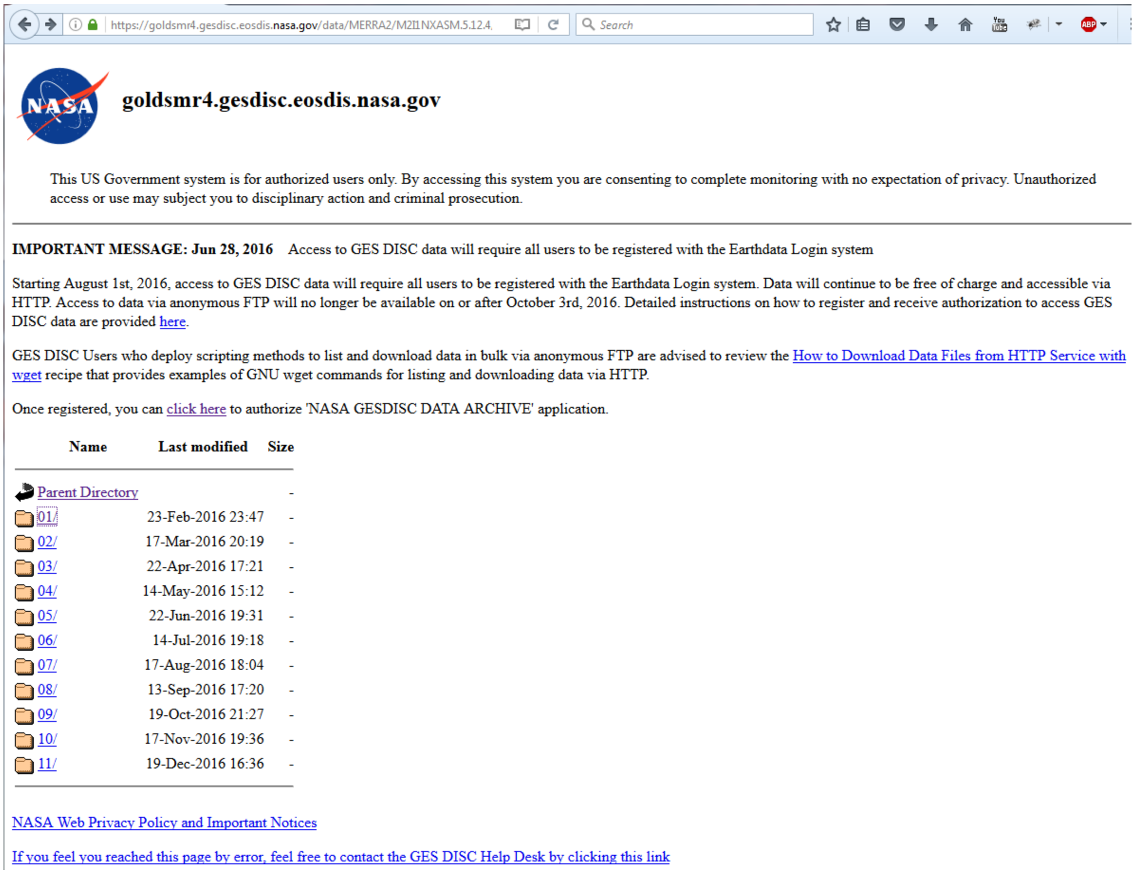


For downloading the 2D surface data, you can use the following link:
https://disc.gsfc.nasa.gov/uui/datasets/M2I1NXASM_V5.12.4/summary?keywords=%22MERRA-2%22&start=1920-01-01

This will take you to the following site:



As before, select the “Online Archive” box, which moves you to the following site, where you can select the dates of interest as above:



https://goldsmr4.gesdisc.eosdis.nasa.gov/data/MERRA2/M2I1NXASM.5.12.4

goldsmr4.gesdisc.eosdis.nasa.gov

This US Government system is for authorized users only. By accessing this system you are consenting to complete monitoring with no expectation of privacy. Unauthorized access or use may subject you to disciplinary action and criminal prosecution.

IMPORTANT MESSAGE: Jun 28, 2016 Access to GES DISC data will require all users to be registered with the Earthdata Login system

Starting August 1st, 2016, access to GES DISC data will require all users to be registered with the Earthdata Login system. Data will continue to be free of charge and accessible via HTTP. Access to data via anonymous FTP will no longer be available on or after October 3rd, 2016. Detailed instructions on how to register and receive authorization to access GES DISC data are provided here.

GES DISC Users who deploy scripting methods to list and download data in bulk via anonymous FTP are advised to review the How to Download Data Files from HTTP Service with wget recipe that provides examples of GNU wget commands for listing and downloading data via HTTP.

Once registered, you can click here to authorize 'NASA GESDISC DATA ARCHIVE' application.

Name	Last modified	Size
Parent Directory		-
01/	23-Feb-2016 23:47	-
02/	17-Mar-2016 20:19	-
03/	22-Apr-2016 17:21	-
04/	14-May-2016 15:12	-
05/	22-Jun-2016 19:31	-
06/	14-Jul-2016 19:18	-
07/	17-Aug-2016 18:04	-
08/	13-Sep-2016 17:20	-
09/	19-Oct-2016 21:27	-
10/	17-Nov-2016 19:36	-
11/	19-Dec-2016 16:36	-

NASA Web Privacy Policy and Important Notices

If you feel you reached this page by error, feel free to contact the GES DISC Help Desk by clicking this link

The following is a partial list of tools for visualizing and manipulating MERRA-2 data specifically or NetCDF data generally:

- Giovanni: https://giovanni.gsfc.nasa.gov/giovanni/
- Integrated Data Viewer (IDV): http://www.unidata.ucar.edu/software/idv/index.html
- Ultrascale Visualization Climate Data Analysis Tools (UV-CDAT): https://uv-cdat.llnl.gov/index.html
- Ferret: http://ferret.pmel.noaa.gov/Ferret/
- NCAR Command Language (NCL): http://www.ncl.ucar.edu/Applications/
- Panoply NetCDF data viewer: http://www.giss.nasa.gov/tools/panoply/download/

Appendix E: Creation of a Default ICAO Type Mapping Table

```
USE [MCM_PDARS_07-04-2014]
GO

--CAUTION: WIPING TABLES CLEAN
--DELETE FROM [dbo].[MVT_FLIGHT_BADA4]
--DELETE FROM [dbo].[MVT_FLIGHT_SRC_LINK]
--DELETE FROM [dbo].[MVT_FLIGHT_ETMS_LINK]

DECLARE @baseActivityDate datetime = '7/4/2014 00:00:00'

--BEGIN CREATION OF DOWN-SELECTED AIRFRAME_ACTYPE_MAP
IF  OBJECT_ID(N'tempdb..#FILTERED_FLT_AIRFRAME_ACTYPE_MAP') IS NOT NULL
DROP TABLE #FILTERED_FLT_AIRFRAME_ACTYPE_MAP

CREATE TABLE #FILTERED_FLT_AIRFRAME_ACTYPE_MAP
(
       [ACTYPE] nchar(10)
       ,[SOURCE] nchar(4)
       ,[AIRFRAME_ID] int
)

INSERT INTO #FILTERED_FLT_AIRFRAME_ACTYPE_MAP
SELECT
       [ACTYPE]
       ,[SOURCE]
       ,MIN([AIRFRAME_ID]) -- RESTRICTS CHOICES TO A SINGLE AIRFRAME_ID; COULD ADD TIME-
DEPENDENCY IF NECESSARY
FROM [MCM_FLEET_FULL_08_19_2016].[dbo].[FLT_AIRFRAME_ACTYPE_MAP] AS map
GROUP BY
       [ACTYPE]
       ,[SOURCE]
ORDER BY [ACTYPE], [SOURCE]

--SELECT * FROM #FILTERED_FLT_AIRFRAME_ACTYPE_MAP --FOR TESTING ONLY

--END CREATION OF DOWN-SELECTED AIRFRAME_ACTYPE_MAP

--BEGIN CREATION OF AIRFRAME-TO-ACCODE MAP
IF  OBJECT_ID(N'tempdb..#AIRFRAME_TO_A_SINGLE_DEFAULT_ACCODE_MAP') IS NOT NULL
DROP TABLE #AIRFRAME_TO_A_SINGLE_DEFAULT_ACCODE_MAP

CREATE TABLE #AIRFRAME_TO_A_SINGLE_DEFAULT_ACCODE_MAP
(
       [AIRFRAME_ID] int
       ,[ACCODE] nvarchar(25)
       PRIMARY KEY([AIRFRAME_ID])
)

INSERT INTO #AIRFRAME_TO_A_SINGLE_DEFAULT_ACCODE_MAP
SELECT
       [AIRFRAME_ID]
       ,[ACCODE]
```

```
FROM
(
	SELECT [ACCODE]
			--,[EFF_DATE]
			--,[EXP_DATE]
			--,[ACNAME]
			,[AIRFRAME_ID]
			,[ACCODES_FOR_A_GIVEN_AIRFRAME_ID] = COUNT([ACCODE]) OVER (PARTITION BY
[AIRFRAME_ID])
		FROM [MCM_FLEET_FULL_08_19_2016].[dbo].[FLT_REF_ACCODES]
		WHERE ACCODE NOT IN ('PA46T', 'CRJ900', 'CRJ700','b727-2fh','b737-2fh','b767-
3f','b777-2f', 'dc9-1fh','crj1000', 'an38-100', 'dc9-3fh') --arbitrary removal
			AND @baseActivityDate BETWEEN EFF_DATE AND EXP_DATE
	GROUP BY
			[ACCODE]
			,[AIRFRAME_ID]
) AS unfiltered_airframe_id_to_accode_map
WHERE [ACCODES_FOR_A_GIVEN_AIRFRAME_ID] = 1

--END CREATION OF AIRFRAME-TO-ACCODE MAP

SELECT
	[ACTYPE]
	,eqp.[EQUIP_ID]
FROM #FILTERED_FLT_AIRFRAME_ACTYPE_MAP AS ac
LEFT OUTER JOIN #AIRFRAME_TO_A_SINGLE_DEFAULT_ACCODE_MAP acc
	on
	(
		acc.[AIRFRAME_ID] = ac.[AIRFRAME_ID]
		--and @baseActivityDate between acc.EFF_DATE and acc.EXP_DATE
	)
LEFT OUTER JOIN
	(
		SELECT
			[EQUIP_ID] = MIN([EQUIP_ID]) --CAUTION: THIS IS ONLY STABLE AS LONG
AS PREEXISTING EQUIPMENT TABLE ENTRIES ARE NEVER CORRECTED (I.E., CHANGES ARE ONLY
APPENDED)
			,[AIRFRAME_ID]
			--, eng.* --FOR TESTING ONLY
		FROM [MCM_FLEET_FULL_08_19_2016].[dbo].[FLT_EQUIPMENT] eqp
		WHERE [BADA4_ID] IS NOT NULL
		GROUP BY
			[AIRFRAME_ID]
		--ORDER BY [EQUIP_ID] --FOR TESTING ONLY
	) one_random_equipment_type_for_bada4_modeling
	ON one_random_equipment_type_for_bada4_modeling.[AIRFRAME_ID] = ac.[AIRFRAME_ID]
LEFT OUTER JOIN
	(
		SELECT
			[EQUIP_ID] = MIN([EQUIP_ID]) --CAUTION: THIS IS ONLY STABLE AS LONG
AS PREEXISTING EQUIPMENT TABLE ENTRIES ARE NEVER CORRECTED (I.E., CHANGES ARE ONLY
APPENDED)
			,[AIRFRAME_ID]
			--, eng.* --FOR TESTING ONLY
		FROM [MCM_FLEET_FULL_08_19_2016].[dbo].[FLT_EQUIPMENT] eqp
		GROUP BY
			[AIRFRAME_ID]
	) one_random_equipment_type
```

```
       ON one_random_equipment_type.[AIRFRAME_ID] = ac.[AIRFRAME_ID]
LEFT JOIN [MCM_FLEET_FULL_08_19_2016].[dbo].[FLT_EQUIPMENT] eqp ON eqp.[EQUIP_ID] =
COALESCE(one_random_equipment_type_for_bada4_modeling.[EQUIP_ID],
one_random_equipment_type.[EQUIP_ID])
WHERE [SOURCE] = 'ICAO' AND ACTYPE != 'HELO'
ORDER BY [ACTYPE]

DROP TABLE #FILTERED_FLT_AIRFRAME_ACTYPE_MAP
DROP TABLE #AIRFRAME_TO_A_SINGLE_DEFAULT_ACCODE_MAP
```

U.S. Department of Transportation
John A. Volpe National Transportation Systems Center
55 Broadway
Cambridge, MA 02142-1093

617-494-2000
www.volpe.dot.gov

DOT-VNTSC-FAA-17-13