IRENE: AE9/AP9/SPM Radiation Environment Model

User's Guide

Version 1.50.001

The IRENE (International Radiation Environment Near Earth): (AE9/AP9/SPM) model was developed by the Air Force Research Laboratory in partnership with MIT Lincoln Laboratory, Aerospace Corporation, Atmospheric and Environmental Research, Incorporated, Los Alamos National Laboratory and Boston College Institute for Scientific Research.

IRENE (AE9/AP9/SPM) development team: Wm. Robert Johnston[1] (PI), T. Paul O'Brien[2] (PI), Gregory Ginet[3] (PI), Stuart Huston[4], Tim Guild[2], Christopher Roth[4], Yi-Jiun Su[1], Rick Quinn[4], Michael Starks[1], Paul Whelan[4], Reiner Friedel[5], Chad Lindstrom[1], Steve Morley[5], and Dan Madden[6].

To contact the IRENE (AE9/AP9/SPM) development team, email ae9ap9@vdl.afrl.af.mil .

The IRENE (AE9/AP9/SPM) model and related information can be obtained from AFRL's Virtual Distributed Laboratory (VDL) website: https://www.vdl.afrl.af.mil/programs/ae9ap9

V1.00.002 release: 05 September 2012

V1.03.001 release: 26 September 2012

V1.04.001 release: 20 March 2013

V1.04.002 release: 20 June 2013

V1.05.001 release: 06 September 2013

V1.20.001 release: 31 July 2014

V1.20.002 release: 13 March 2015

V1.20.003 release: 15 April 2015

V1.20.004 release: 28 September 2015

V1.30.001 release: 25 January 2016

V1.35.001 release: 03 January 2017

V1.50.001 release: 01 December 2017

Document Author: Christopher Roth/AER

[1] Air Force Research Laboratory, Space Vehicles Directorate
[2] Aerospace Corporation
[3] MIT Lincoln Laboratory
[4] Atmospheric and Environmental Research, Incorporated
[5] Los Alamos National Laboratory
[6] Boston College Institute for Scientific Research

Table of Contents

Acknowledgements

The IRENE (AE9/AP9/SPM) model development team thanks the MagEIS, REPT and RPS sensor groups of the Van Allen Probes satellite team, and the HiLET sensor team, especially S. Claudepierre, R. Selesnick, J. Mazur and D. Turner, for their help preparing the data for inclusion in v1.5.

The team also thanks D. Heynderickx, E. Daly, and the European Space Agency for providing the cleaned data from the Azur satellite.

IRENE (AE9/AP9/SPM) Model Software Suite Overview

The International Radiation Environment Near Earth (IRENE) radiation belt and space plasma specification model software suite provides estimates of trapped energetic electrons, energetic protons, and plasma from the AE9, AP9 and SPM models, for use in space system design, mission planning, and other applications of climatological specification. It is based on 45 satellite-based data sets processed to create maps of the particle fluxes along with estimates of uncertainties from both imperfect measurements and space weather variability. These uncertainty estimates can be obtained as statistical confidence levels, e.g., the 50th and 95th percent, for fluxes and derived quantities, supporting design trades. Implementations of the legacy AE8/AP8 and CRRESELE/CRRESPRO models are available within the model suite. The SHIELDOSE2 code is used for dose estimation from the calculated fluxes. Orbit ephemeris may be generated using one of three orbit propagators or may be directly supplied. The self-contained software package includes a Windows-based executables, accessible either by command line or graphical user interface, plus supporting documentation such as the users' guide, validation results, and software license information.

The IRENE model package is hosted at https://www.vdl.afrl.af.mil/programs/ae9ap9. This public site includes both general and detailed information about the AE9/AP9/SPM model collection as well as instructions for obtaining the current model software. The model package is publicly available at no cost and can be downloaded after registering on the website with contact information. For platforms other than Windows, the source code version must be directly requested. Build instructions are included for Linux platforms.

The IRENE model package continues to be actively developed by a collaboration led by the Air Force Research Laboratory (AFRL) and including Aerospace Corporation, Atmospheric and Environmental Research, Inc., Los Alamos National Laboratory, and Massachusetts Institute of Technology Lincoln Laboratory. Contact the development team at ae9ap9@vdl.afrl.af.mil.

This IRENE software distribution contains all files needed to install and execute the C++-based radiation environment models, using a command-line application, or through a graphical user interface (GUI). An Application Programming Interface (API) is also included in the model distribution, for integrating the model calls into user applications.

The command-line application CmdLineAe9Ap9 loads the model parameters and orbital position specifications from user-constructed input files, and produces the corresponding set of output files containing the ephemeris, flux, fluence, and/or dose calculation results, as requested.

The GUI application Ae9Ap9Gui provides a graphical front-end to the CmdLineAe9Ap9 application. The orbital path and the various model settings and parameters may be easily specified in a user-friendly format. The CmdLineAe9Ap9 application is then executed automatically, using the input files generated according to the user's selections. Basic 2D plots of the various model results may also be produced.

Model Software Code Stack

<table>
<tr><td colspan="6">Graphical User Interface Application
User-friendly access for execution of AE9/AP9 and other models, with basic graphical outputs</td></tr>
<tr><td colspan="6">Command-Line Application
Input-file driven execution of AE9/AP9 and other models; can be run in batch mode, or set up with scripts</td></tr>
<tr><td colspan="6">Application-Level API
• C++ interface for specification of model run parameters, output file generation and access to results
• Features parallelized processing capabilities for most types of calculations
• Ephemeris generation using choice of orbit propagators
• Several modes for flux calculations using AE9/AP9/SPM models
• Flux calculations using legacy models (e.g. AP8/AE8, CRRES)
• Accumulates flux results over time in multiple modes (cumulative, interval, full, boxcar, exponential)
• Calculates dose rate and accumulated dose values from fluxes, for specified thicknesses (via ShieldDose2)
• Aggregates flux/fluence/dose results from sets of ‘Perturbed Mean’ or ‘Monte Carlo’ scenario outputs</td></tr>
<tr><td colspan="6">Scheduler
Determines model processing units needed for performing requested calculation, and coordinates parallelization</td></tr>
<tr><td colspan="6">Model-Level API
C++ interfaces to each of the component model processing units</td></tr>
<tr><td>Ephemeris</td><td>Ae9Ap9</td><td>Accumulation</td><td>Dose</td><td>Aggregation</td><td>Legacy</td></tr>
<tr><td colspan="6">Supporting Models
• Orbit Propagation
• Magnetic Field and coordinate conversions
• Adiabatic Invariant Coordinates
• ShieldDose2
• Legacy Radiation Belt models</td></tr>
<tr><td colspan="6">Lower-level Utilities
• Database access utilities
• Boost and HDF5 third-party libraries
• system libraries</td></tr>
</table>

In addition to this User’s Guide, there are several other files of information located in the ‘`Ae9Ap9/documents`’ directory. These provide a detailed overview of the IRENE package of models (AE9/AP9/SPM), as well as its validation and comparisons. The Application Programming Interface (API) documentation and source build instructions are also included here, as well as a set of license information files.

Several annotated sample input files for use with the CmdLineAe9Ap9 application may be found in the ‘`Ae9Ap9/samples`’ directory, along with their associated ‘expected output’ files, for verifying proper model installation and operation.

The ‘`Ae9Ap9/unitTests`’ directory contains many more input files, and their associated ‘expected output’ files, used in performing much more extensive testing and validation of the model software.

Model Installation

Version 1.50.001 of the IRENE (AE9/AP9/SPM) model package is distributed as a zipfile, '`Ae9Ap9_version_1.50.001.zip`', containing Windows release-mode binaries and libraries, model databases, sample input and output files and supporting documentation files. The separate source distribution may be obtained from the model development team upon request, for generating custom applications using the API methods, or for building the software executables on Linux platforms.

Important: If a previous version of the Ae9Ap9 software is present in the directory where this new version is to be installed, please remove or rename it prior to unzipping the distribution file. The distribution has the directory structure shown below:

```
Ae9Ap9
   ├── bin
   │     ├── win32
   │     └── win64
   ├── documents
   │     └── Licenses
   ├── modelData
   ├── samples
   │     └── expectedOutput
   ├── source
   └── unitTests
         └── expectedOutput
```

For a Windows installation, all that is required is to unzip the distribution file.
For a Linux-based installation, unzip the *source distribution* file (available upon request), then refer to the detailed instructions in the '`Build_Instructions_for_AE9AP9.pdf`' file in the '`Ae9Ap9/documents`' directory.

Installation Testing

Open a command prompt window, navigate to the directory containing the binary executables (eg '`Ae9Ap9/bin/win64`'), enter the command:

```
CmdLineAe9Ap9 –i ../../samples/Ap9ShortInput.txt
```

The resulting output files will be written to this '`samples`' directory. Verify that the number of files generated, and respective file contents match that of the "`Ap9ShortOutput`"-prefixed files located in the '`../../samples/expectedOutput`' directory. Other test input files are also available in this '`samples`' directory, and their corresponding output files in the '`expectedOutput`' subdirectory. See Appendix H for if any problems are encountered.

The GUI application may be started by entering `Ae9Ap9Gui` in that same command line window, or double-click the application's icon in the Windows Explorer window for the directory. A detailed description of the GUI application follows in a later section of this document.

Once tested, see Appendix G for tips to customize the model software installation.

There are many executable files present in the `Ae9Ap9/bin/win`*XX* (or `linux`) directory. Only the CmdLineAe9Ap9 and Ae9Ap9Gui applications, and the TotalDose, IntegralPlasma and ConvertToXlsx post-processing utility applications (described later) are directly usable at the command line. All other executable files are 'helper' applications that are used internally to support the various operations performed in the full model calculation process.

In the `Ae9Ap9/bin/win64` directory, there are several additional files that comprise the 'Runtime Environment' of the Intel MPI Library. When executing on a Windows *cluster*, an extra argument may be required: '`-b`' specifies the use of SSH for the MPI communication; '`-c`' specifies the use of the Intel 'hydra_service' utility (separately invoked). See https://software.intel.com/en-us/node/528873 for more information about this utility.

Important: Some security suites installed on Windows machines may contain 'program control' features. Because many of these 'helper' applications are executed internally by the other main applications, it is recommended that the applications in this directory are pre-configured in the security suite to be 'allowed' or 'trusted' before running the Ae9Ap9 software. It is possible that the security suite prompts (asking the user if their execution is allowed) could disrupt the internal communication between them, causing the model run calculations to 'hang'.
See Appendix H for more details.

CmdLineAe9A9 Application Quick Reference

`CmdLineAe9Ap9 <`*`args`*`>`

`-h`	lists these program argument options, and exits
`-v`	reports model/software version information, and exits
`-i <`*`filename`*`>`	specifies model run input file to process
`-n <`*`numproc`*`>`	specifies number of processors to use (defaults to 1) - includes one processor for master when multi-threaded - overrides NumProc parameter in input file, if present - specify a *negative* number for use on cluster† systems, or to bypass the processor hardware query/limitation
----- the following apply to Win64 multi-threaded operation (uses Intel MPI Library)* ------	
<*no extra arg*>	MPI communication for a single multi-processor machine, or a cluster that acts like a single machine (not clusters)
`-b`	uses 'SSH' for MPI communication (single and clusters)
`-c`	requires active 'hydra_service' utility (single and clusters)

† Cluster usage may require 'reservation' or 'allocation' of processors
* Multi-threaded operation on Win32 platforms is *not* supported

Command-Line Application

The CmdLineAe9Ap9 application is driven by a parameter input file for running the 'AE9/AP9' and 'Plasma' models at time-tagged orbital positions, such as satellite ephemeris or a ***grid*** of coordinates. The requested model calculation results are written to ASCII text files, in comma-separated-value format by default; several input and output formatting options are available. Other 'legacy' radiation belt models, 'AE8', 'AP8', 'CRRESELE', 'CRRESPRO' and 'CAMMICE/MICS', are also available within this same application.

The CmdLineAe9Ap9 application reads parameter settings from an external input file, specified by the command-line argument '`-i <filename>`'. Parallel processing is available* for the 'AE9/AP9' and 'Plasma' model calculations; the number of processors to be used may be specified with an additional argument '`-n <numproc>`', where `<numproc>` can be 3 or larger, up to the maximum number of processing nodes available (note that this count *includes* one node to be the 'master') on the host machine or cluster. The number of processors specified on the command line will always override the 'NumProc' specification (if present) within the input file. The application will query the *local* host system to determine the number of processors that are present; the number is capped at this limit. However, for *cluster systems*, or cases where this hardware detection fails, or provides incorrect information, this query can be disabled by specifying a ***negative*** number. Please note that specifying more processors than actually present will impact the application execution performance. For cluster systems, a 'reservation' or 'allocation' of processing nodes may be also required before their use – check with your local cluster administrator for details. Execution on a *Windows cluster* may require the use of the Intel MPI "hydra_service" utility (included in the `bin/win64` directory) on each node, and the additional '`-c`' argument on the CmdLineAe9Ap9 command line.

Its use of an input file enables the CmdLineAe9Ap9 application to be used in "batch" mode or within a script, and/or distributing large modeling tasks across multiple processors and servers. The model run times can also be greatly reduced by taking advantage of the parallel-processing capabilities when the appropriate hardware resources are available.

The input file being supplied to the CmdLineAe9Ap9 application is accessed multiple times during the model calculation processing. Therefore, it is crucial that this file is *not modified* while the application is being executed, otherwise the model run may fail and/or produce incorrect results.

The construction of the input file used to drive the CmdLineAe9Ap9 application, as well as the details about each of the available parameters, is described in the following sections. Several working sample input files, such as '`Ax9InputSample.txt`', are provided in the '`Ae9Ap9/samples`' directory.

* Parallel processing is NOT supported for ***32-bit*** Windows platforms.

Input File Construction

The basic format of the input file is 'keyword/value' pairs on each line: the parameter keyword name, followed by a colon and then the value (or set of comma-separated values) for the parameter. Keywords and string values are *not* case-sensitive. Keywords or string values shown in italics in the tables below are deprecated, but still accepted in this software release. No quotes are needed for string values. Input file lines that begin with a '#' symbol are treated as comments, and are therefore ignored by the application. The ordering of the parameter specification lines in the input file does not matter. Any *relative* paths specified for files or directories should be relative to the working directory from which the CmdLineAe9Ap9 application is to be executed.

CmdLineAe9Ap9 input file settings can be logically grouped into the following categories:

Basic Model Inputs	– core model parameters, required for any model run
Advanced Model Inputs	– optional settings for input/output formatting and pitch angles
Accumulation and Aggregation Inputs	– optional settings for combining complex output results
Dose Calculation Inputs	– optional settings to drive the ShieldDose2 model calculations
Orbit Propagator Inputs	– optional settings for generating orbit ephemeris

Input file settings for each of these categories are described in the following tables.

Any of the parameter values requiring a path and/or filename may be specified using an environment variable; ie `MagFieldDB: $AX9HOME/modelData/igrfDB.h5` , provided the '`AX9HOME`' environment variable has been previously defined (external to the CmdLineAe9Ap9 application). Environment variables may be referenced using either the Windows (`%AX9HOME%`) or Linux (`$AX9HOME`) form.

Within these following tables, when more than one parameter keyword is shown, the first one is preferred. The keywords and values that are marked as 'deprecated' were used in prior versions of the software; their continued use is strongly discouraged.

Parameter values separated with a '|' symbol (ie, 'millimeters | mm') are values that are equivalent; either one may be used.

Descriptions for the parameters used with the "Legacy" models (AE8, AP9, CRRESELE, CRRESPRO and CAMMICE/MICS) may be found in Appendices A and B.

Basic Model Inputs

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
ModelType	AE9 AP9 PLASMA *[valid for $2 \le L_m \le 10$]* Legacy models*: AE8, AP8, CRRESELE, CRRESPRO, CAMMICE	Required	None	Type of model to be run (requires corresponding database file specified in ModelDB parameter). Calculated model results will be *omnidirectional* flux values, unless pitch angles or direction vectors are specified. *See Appendices A and B for Legacy model parameters
ModelDB	*<path>*†/AE9V15_runtime_tables.mat* *<path>*/AP9V15_runtime_tables.mat *<path>*/SPMEV12_runtime_tables.mat *<path>*/SPMHV12_runtime_tables.mat *<path>*/SPMHEV10_runtime_tables.mat *<path>*/SPMOV10_runtime_tables.mat †<path> may be specified using an environment variable	Required	None	Database file used to drive the model, corresponding to the selected ModelType). Must include path to file (absolute, or relative to the location from which the CmdLineAe9Ap9 application is executed). For PLASMA: specify the appropriate 'SPM*' species file: 'E' (electrons), 'H', 'HE' (He), 'O'; *Use the database files included with the model distribution.
MagFieldDB	*<path>*†/igrfDB.h5	Required	None	Database file for the magnetic field model, including path
KPhiNNetDB	*<path>*†/fastPhi_net.mat	Required	None	Database file for the K/Phi coordinate neural network, including path
KHminNNetDB	*<path>*†/fast_hmin_net.mat	Required	None	Database file for the K/Hmin coordinate neural network, including path
OutFile	valid path† and file name prefix *ie 'Run/Run1.AE9.output.txt'* see page *36* for more details.	Required	None	A path and filename "prefix" that will be used when generating the model output files; the naming of these files is based on this prefix and the various model data output and aggregation parameters also specified in the input file. **Any previously-generated output files with this same "prefix" will be *overwritten* ("Leftovers" may also still exist).**

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
OrbitFile	valid path† and file name of ephemeris file †path may be specified using an environment variable This input "ephemeris" file can alternatively contain a *grid* of coordinate positions, for a fixed date and time	Required	None	File containing time and position information (ephemeris) that is used in one of two ways: •*Output*: when orbit definition and propagation parameters are also specified, ephemeris is generated and written to this file. Any existing file will be *overwritten* with this new ephemeris. Formatting of this ephemeris file may be controlled with the TimeSpec, CoordSys, CoordUnits, CoordOrder and DataDelim parameters. See the 'Advanced Model Inputs' table. •*Input*: ephemeris or grid is read from this existing file. If necessary, formatting of this ephemeris/grid file may be described through the use of the InTimeSpec, InCoordSys, InCoordUnits, InCoordOrder and InDataDelim parameters.
FluxType	1PtDiff 2PtDiff* Integral† ('Diff' and 'Differential' = '1PtDiff') [Only '1PtDiff' can be used for dose calculations]	Required	None	Type of flux to be computed. *Two-point differential requires 'Energies' *and* 'Energies2' parameter values. †For Plasma model *integral* flux, see 'IntPlasma_Readme.txt' for details.
Energies	AE9: 0.04 – 10.0 (MeV) AP9: 0.1 – 2000.0 (MeV) *[Plasma models valid for $2 \le L_m \le 10$]* Plasma/electrons: 0.001 – 0.040 (MeV) Plasma/ions: 0.00115 – 0.1643 (MeV)	Required	None	Comma-separated list of energy levels, in MeV, at which flux values are to be computed, at each time step. See Appendix D. Energy values are restricted to their model-specific ranges. When the Plasma model is used in tandem with the Ax9 model, it is recommended to use electron energies <0.04 MeV, and protons <0.1 MeV.
Energies2	(same ranges as 'Energies') For Plasma model *integral* flux, see 'IntPlasma_Readme.txt' for details.	Optional	None	Used only when 'FluxType' = '2PtDiff'. A comma-separated list of energy levels that define the *end* of the energy ranges, between which fluxes are computed. The 'Energies' parameter defines the *start* of each range.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
FluxOut	Mean Percentile, ## Perturbed, ### MonteCarlo, ###* Multiple comma-separated numbers may be specified on the same line, as well as ranges (ie "perturbed,1,2,5-10,20")	Required	None	Flux data to be generated/output. This parameter may appear multiple times. The 'mean' is a simple mean value. The 'percentile' numbers may be in the range 1-99. The 'perturbed' mean and/or 'montecarlo' scenario identification numbers may be in the range 1-999. See the full explanation in the 'Flux Data Mode Details' on page 16. *montecarlo is not applicable for 'Plasma' model
FluenceOut FlueOut	True False Deprecated*: mean, percentile, perturbed, montecarlo*: interpreted as 'True'	Optional	False	Fluence data to be generated/output. This parameter follows all values specified the 'FluxOut'. Specification of any of the 'Accumulation' parameters (described later) may affect the fluence results produced.
DoseRateOut* DoseOut*	True False Deprecated*: mean, percentile, perturbed, montecarlo*: interpreted as 'True'	Optional	False	Dose rate data to be generated/output. This parameter follows all values specified the 'FluxOut'. For use with *omnidirectional 1PtDiff flux only*. Specification of any of the 'Accumulation' parameters (described later) may affect the dose rate results produced. *Requires additional Dose Calculation input parameters, in a following section. Not applicable for 'Plasma' model.
DoseAccumOut* CDoseOut*	True False Deprecated*: mean, percentile, perturbed, montecarlo*: interpreted as 'True'	Optional	False	Cumulative dose data to be generated/output. This parameter follows all values specified the 'FluxOut'. For use with *omnidirectional 1PtDiff flux only*. Specification of any of the 'Accumulation' parameters (described later) may affect the cumulative dose results produced. *Requires additional Dose Calculation input parameters, in a following section. Not applicable for 'Plasma' model.

Supported Coordinate Systems

CoordSys	Full Coordinate System Name	Default Coordinate Values(units)
GEI	Geocentric Earth Inertial or Earth-Centered Inertial (ECI)	X(Re), Y(Re), Z(Re)
GEO	Geocentric Cartesian	X(Re), Y(Re), Z(Re)
GDZ	Geodetic (WGS84)	Altitude(km), Lat(deg), Lon(deg)*
GSM	Geocentric Solar Magnetospheric	X(Re), Y(Re), Z(Re)
GSE	Geocentric Solar Ecliptic	X(Re), Y(Re), Z(Re)
SM	Solar Magnetic	X(Re), Y(Re), Z(Re)
MAG	Magnetic (tilted dipole)	X(Re), Y(Re), Z(Re)
SPH	Spherical	Radius(Re), Colatitude(deg), Lon(deg)*
RLL	Radius, Latitude, Longitude	Radius(Re), Lat(deg), Lon(deg)*
1 Re = 6371.2 km		*East Longitude
Latitude = -90° – +90°; Colatitude = 0° – 180°; Longitude = -180° – +360°		
See [Bhavnani and Vancour, 1991] for full descriptions of these coordinate systems.		

Supported Time Specifications

TimeSpec	Time Field Components
MJD	Modified Julian Date (Ref = 17 Nov 1858 0000GMT, See Appendix C)
YrDDDFrac	Year, DayOfYear+fraction
YrDDDGmt	Year, DayOfYear, Gmtsec
YrMoDaGmt	Year, Month, Day, Gmtsec
YrMoDaHrMnSc	Year, Month, Day, Hour, Minute, Seconds
Modified Julian Date = 33282.0 – 69806.99999	
Year = 1950 – 2049 (four digits)	
DayOfYear = 1 – 365/366 (+fraction= .00000-.99999)	
Month = 1 – 12; Day = 1 – 28/29/30/31	
Gmtsec = 0 – 86399.99	
Hour = 0 – 23; Minute = 0 – 59; Seconds = 0 – 59.99	

Flux Data Mode Details

The 'FluxOut', and associated parameters, is used specify the various modes of flux data to be returned by the model. The 'mean' and 'percentile' modes capture the statistical behavior of the data upon which the model was built. The 'perturbed' mode adds the uncertainties in the mean flux maps that are due to measurement and gap-filling errors. The 'montecarlo' mode contains all these uncertainties of the 'perturbed' mode, then adds an estimate of the dynamic variations due to space weather processes. The multiple "scenarios" of the latter two modes are specified with scenario identification numbers (used to produce a random number seed), each producing a different flux profile for every orbit, bounded by the variances due to measurement error and space weather. These variations represent the range associated with space weather on multiple timescales and span the variability observed throughout a solar cycle; however, the solar cycle phase is not reproduced. This scenario "seed" number enables these results of the model calculations to be fully reproducible, provided that the same ephemeris information and time reference ('MCEpoch', described in the next section) are used. Unless otherwise specified, this time reference defaults to the first time of the ephemeris information.

The 'MCEpoch' parameter may also be used for breaking up long-duration 'montecarlo'-mode simulations into separate model runs. Each time segment can be run separately and then manually combined afterwards, provided that the same 'MCEpoch' value is specified for all time segment model runs. Such processing is now done automatically when the parallelized execution mode is used.

By aggregating the results of a large number of mission scenarios (each with a different seed number), the confidence levels of any quantity derivable from the flux spectrum, e.g. fluence (time integrated flux) or accumulated dose, may be calculated in terms of probabilities of occurrence during the course of the mission or on other timescales.
Much more information about the AE9, AP9 and SPM models may be found in [Ginet, et al, 2013] ('*AE9AP9SPM_SSR_Overview*'), and the VDL website (see page 2). Also see the "*Known Issues and Limitations*" document included in the model distribution files.

Differences in AE9/AP9/SPM Run Modes Affecting the Calculated Flux Results

Run Mode	Internal Model Flux Map Value Initialization	Internal Model Flux Map Behavior during Run	Model Flux Outputs	Notes on Uses of Flux Output
Mean or Percentile	Mean or Percentile values	Remains static throughout	Mean or selected Percentile	Mean provides a quick estimate of the environment and effects. Percentiles are appropriate only for comparing with measurements at a given location and energy
Perturbed Mean	Mean values with random perturbations (for each scenario)	Remains static throughout	Multiple scenarios provide confidence intervals based on model uncertainties	Perturbed Mean is appropriate for cumulative/integrated quantities (e.g., fluence, Total Ionizing Dose [TID])
Monte Carlo	Mean values with random perturbations (for each scenario)	Evolves as time progresses	Multiple scenarios provide confidence intervals including space weather (e.g., worst-case over specified time intervals)	Monte Carlo provides an estimate of uncertainty in time-varying quantities (e.g., Single Event Effect [SEE] rates, deep dielectric charging)

Choices of AE9/AP9/SPM Model Runs Suggested for Typical Specifications

Specification Desired	Run Mode	Duration	Combination of Results	Notes
Total Dose	Perturbed Mean	Several orbits or days	SPME + AE9, SPMH + AP9 + Solar*	*Solar Protons: see [Xapsos, et al, 2013]
Displacement Damage (proton fluence)	Perturbed Mean	Several orbits or days	AP9 + Solar*	
Proton SEE (proton worst case)†	Monte Carlo	Full mission	AP9 + Solar*	†Do not include SPM*x* models in 'worst case' runs
Internal Charging (electron worst case)†	Monte Carlo	Full mission	AE9	

- Run 40 scenarios through either static Perturbed Mean or dynamic Monte Carlo
- Compute statistics by comparing results across scenarios (e.g., in what fraction of scenarios does the design succeed)

See [O'Brien, 2014] for more information on performing AE9/AP9/SPM model runs for desired specifications.

Advanced Model Inputs

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
TimeSpec	MJD YrDDDFrac YrDDDGmt YrMoDaGmt YrMoDaHrMnSc	Optional	MJD [Modified Julian Date]	Format of time specification to be used for the model output files and generated [Output] OrbitFile. See the 'Supported Time Specifications' table on page 16. See Appendix C for more details about Modified Julian Dates.
InTimeSpec	MJD YrDDDFrac YrDDDGmt YrMoDaGmt YrMoDaHrMnSc	Optional	*(value for TimeSpec)*	Format of time specification contained in the specified [Input] OrbitFile. See the 'Supported Time Specifications' table on page 16. See Appendix C for more details about Modified Julian Dates.
CoordSys	GEI GEO GDZ GSM GSE SM MAG SPH RLL	Optional	GEI [Geocentric Earth Inertial]	Coordinate system of position values that are written to the model output files and generated [Output] OrbitFile. See the 'Supported Coordinate Systems' table on page 16.
InCoordSys	GEI GEO GDZ GSM GSE SM MAG SPH RLL	Optional	*(value for CoordSys)*	Coordinate system of position values that are read from the specified [Input] OrbitFile (and optional DirFile or PitchFile) See the 'Supported Coordinate Systems' table on page 16.
CoordUnits	Re km	Optional	Re [Earth Radii] km for 'GDZ'	Distance units to be used for the Cartesian coordinates, radius or altitude values in the model output files and generated [Output] OrbitFile. 1 Re = 6371.2 km.
InCoordUnits	Re km	Optional	*(value for CoordUnits)*	Distance units of the Cartesian coordinates, radius or altitude values contained in the specified [Input] OrbitFile.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
CoordOrder	Standard | Std Inverted | Inv	Optional	Standard	Order of *non-Cartesian* coordinates to be used for the model output files and generated [Output] OrbitFile. [ignored for Cartesian CoordSys types] 'Standard' = Distance,Lat,Lon 'Inverted' = Lat,Lon,Distance
InCoordOrder	Standard | Std Inverted | Inv	Optional	*(value for CoordOrder)*	Order of *non-Cartesian* coordinates contained in the specified [Input] OrbitFile. [ignored for Cartesian InCoordSys types] 'Standard' = Distance,Lat,Lon 'Inverted' = Lat,Lon,Distance
DataDelim	Comma Space Tab	Optional	Comma	Data delimiter character to be used for the model output files and generated [Output] OrbitFile.
InDataDelim	Comma Space Tab	Optional	*(value for DataDelim)*	Data delimiter character contained in the specified [Input] OrbitFile.
PitchAngles‡ ‡Only one of these direction specifications may be used.	List of comma-separated pitch angle values, in range of 0 – 180 (degrees) [*Cannot be used with Dose calculations*]	Optional	None	Used for generating *unidirectional* model flux results. List of one or more pitch angle values to be used at *all* ephemeris positions in the specified OrbitFile.
PitchFile‡	valid path† and filename of pitch file [*Cannot be used with Dose calculations*] †path may be specified using an environment variable	Optional	None	Used for generating *unidirectional* model flux results. Data file containing one or more pitch angles associated with each of the ephemeris positions in the specified OrbitFile.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
DirFile‡ ‡Only one of these direction specifications may be used.	valid path† and file name of direction file [*Cannot be used with Dose calculations*] †path may be specified using an environment variable	Optional	None	Used for generating *unidirectional* model flux results. Data file containing one or more sets of direction vectors associated with the each of the ephemeris positions in the [Input or Output] OrbitFile. The coordinate system (specified by InCoordSys or CoordSys) *must* be one of the *Cartesian* coordinate systems for proper application.
AdiabatOut *OutputAdiabat* *(deprecated)*	True False	Optional	False	Generate secondary ephemeris output file (named in the form '<OutFile>`_adiabat.txt`') containing geomagnetic parameter and adiabatic invariant coordinate values (MLT, B_{local}, $B_{equator}$, L_m, L*, Phi, K, H_{min}, $Alpha_{eq}$), as a function of pitch angle. These values are calculated and used internally by the model routines. See Appendix D and E for more information. For omnidirectional model flux calculations, these values are reported for the 90° pitch angle only; for unidirectional, they are reported for each derived or specified pitch angle.
MCEpoch *Epoch* *(deprecated)*	Time value, in one of the supported time specification forms*.	Optional	First ephemeris point time	Specifies the reference time for "Monte Carlo" scenario perturbation calculations. This long-period MC model runs to be broken up over time. See the 'Flux Data Mode Details' on page 16 for more information. *See the 'Supported Time Specifications' table on page 16. Values are comma-separated, when requiring multiple fields.
FluxPert	enable disable	Optional	enable	Enable or disable the flux perturbations in "Monte Carlo" scenarios. Setting this parameter to 'disable' is generally only useful for validation or where perturbations dwarf physical features of interest.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
ChunkSize *PtsPerCall* *(deprecated)*	>60 (recommended)	Optional	240	Model performance tuning parameter; specifies the number of ephemeris points to be used in each 'chunk' for the internal model processing. This number directly relates to amount of memory used for storing results during processing, and inversely to frequency of updates of data processing progress. For computer systems with limited memory, performance can be improved by specifying a lower number, such as 120. Due to the overhead required for the processing of each 'chunk', performance will *degrade* with very low numbers.
NumProc* **not* supported on *Win32* platforms	1 (for single-threaded operation) 3 or greater (parallelized operation) <u>For operation on a cluster:</u> A 'reservation' or 'allocation' of nodes/processors may be required. Consult your local cluster system administrator and/or user guide. The hardware system query may not accurately determine the number of processors – <u>specify a *negative* number</u> to bypass this query.	Optional	1	Total number of processors to employ for the model run. For parallelized operation, this number specified *includes* one for the 'master' control node. Alternatively, the number of processors may be specified as a command line argument (which will override this parameter). A hardware system query will be performed to determine the number of existing processors, and the number will be *reduced* to this limit, if exceeded. Specifying the value as a negative number will bypass the hardware query, enabling any number to be used. However, parallel performance will be degraded if insufficient hardware resources are available.
NumFileIo	1 or greater, up to |NumProc|-1	Optional	1	Number of processors to employ during the intense file I/O steps of a multi-threaded model run. Use of this parameter is recommended only when using <u>*RAID-5* *disk units*</u>; otherwise performance may be degraded.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
AsciiOutput	All None ----- or ----- Ephem, Flux, Fluence | Flue, DoseRate | Dose, DoseAccum | CDose, Aggreg, Adiabat multiple comma-separated values may be specified on the same line	Optional	All	Controls the generation of ASCII-formatted files from their intermediate binary output files produced during the model processing steps. When not 'All' or 'None', the ASCII files will be generated only those data types specified*. This parameter may appear multiple times. * the exclusion of certain types of ASCII files may prevent the proper operation of the IntegralPlasma or TotalDose post-processing utilities.
WorkDir	valid path† to readable and writable directory When using multiple processing nodes, this location must be read/write accessible by *all* nodes †path may be specified using an environment variable	Optional	<directory of OutFile>	A working directory where intermediate model output files are to be stored. This will be created if it does not exist. During the model processing, a uniquely-named subdirectory of 'WorkDir' will be created here for the binary-format model output files. Unless specified otherwise, the *sub*directory is removed at the completion of the model calculations. Use of this parameter may improve model run performance if the read/write speeds from/to this location are faster than the default, due to hardware capabilities and/or network configuration. Also see the 'NumFileIo' parameter.
BinDirName	Name of binary file subdirectory (name only – do not include path; may contain _-.+= special characters)	Optional	<unique name derived from OutFile and PID>	Name of binary file subdirectory to be created in the WorkDir directory. Use of parameter is only needed for diagnostic purposes or the retention of the binary files for external uses.
DelBinDir	True False	Optional	True	Controls the automatic deletion of the binary file subdirectory at the completion of the model processing. Setting this parameter to 'False' is only needed for diagnostic purposes, or the retention and/or direct access to the binary files.

Accumulation and Aggregation Inputs

Accumulation (for summation or averaging) is performed over a specified time period (an 'interval') on the data values within each output file. These settings will affect the results for fluence, dose rate and accumulated dose calculations. Several modes of accumulation are available, and multiple modes may be used within a single model run. Aggregation is used when calculating the statistical confidence level values of the like data values amongst a set of output scenario files, such as those generated from 'FluxOut/perturbed' and/or 'FluxOut/montecarlo' settings; these results are statistically meaningful only with at least ten scenarios.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
AccumMode	Cumul Intv | Interval Full Boxcar* Expon* | Exponential* multiple comma-separated values may be specified on the same line * The Boxcar and Exponential accumulation modes are currently considered "experimental"	Optional	Interval	Accumulation mode in which the flux values are processed and the results reported. This parameter may appear multiple times. • *Cumul* – cumulative: data calculations are performed at each time step of ephemeris. Warning: if dose results are requested, note that their calculations are relatively slow; when done at each time step, the overall model run performance will be impacted. • *Interval* – reports data accumulation results at fixed intervals (specified by AccumInterval[Sec]. At the end of processing, any remaining (partially-completed) interval results are also reported. • *Full* – accumulation of values during the full time period covered by the ephemeris, reporting only a single set of results at the end. • *Boxcar** – flux averaging within overlapping windows of time intervals, its length is specified by AccumInterval[Sec], and its advancement in time is specified by AccumIncrem<Sec|Frac>. • *Expon** – performs exponential flux averaging, results are reported at time intervals, specified by AccumInterval[Sec].
AccumInterval *DoseIntrvl* *(deprecated)*	> 0 (units of days)	Optional	1.0	Accumulation interval (in days) for the *Interval*, *Boxcar* and/or *Exponential* modes over which to calculate fluence, average flux, and (*Interval* only) average dose rate and/or dose accumulation.
AccumIntervalSec	> 0 (units of seconds)	Optional	86400	Same as AccumInterval, except in units of seconds

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
AccumIncremSec†	> 0 (units of seconds) but less than *AccumInterval* length	One of these is required for *Boxcar*	None	Used with AccumMode 'Boxcar' mode. Specifies the time advance, in seconds, for start of subsequent interval windows.
AccumIncremFrac† †Only one of these time specifications may be used.	Between 0.0 and 1.0 (exclusive)		None	Used with AccumMode 'Boxcar' mode. Specifies the time advance, expressed as a fraction of AccumInterval[Sec] length, for start of subsequent interval windows.
Aggregate Deprecated: *PMAggX, PMAggF, PMAggD, PMAggCD, MonteAggX, MonteAggF, MonteAggD, MonteAggCD*	m*ean** Median Percent, ## [Multiple comma-separated numbers may be specified on the same line, as well as ranges (ie "percent,50,75,90-99")]	Optional	None	Aggregate the Flux, Fluence, DoseRate and/or DoseAccum output results across 'perturbed mean' and/or 'montecarlo' scenarios defined in the FluxOut settings, computing the confidence levels for the specified percent (0-100) value at each data entry. *See the paragraph below for important details.* The 'median' is the same as 50 percent. This parameter may appear multiple times. * The 'mean' of an aggregation has an indeterminate meaning; its usage is NOT recommended.

The values of the aggregation confidence levels are calculated using the percentile calculation method recommended by the National Institute of Standards and Technology (NIST). The endpoints of 0 and100 percent levels are excluded, as well as additional neighboring levels when fewer than 100 scenarios are used in the aggregation. The 0 percent level returns the lowest scenario value; the 100 percent level returns the highest scenario value.

$$x = f(p,N) = \left\{ \begin{array}{c} 1,\ p \in \left[0, \frac{1}{N+1}\right] \\ p(N+1),\ p \in \left(\frac{1}{N+1}, \frac{N}{N+1}\right) \\ N,\ p \in \left[\frac{N}{N+1}, 1\right] \end{array} \right\} \begin{array}{l} \text{Lowest scenario value} \\ \text{Confidence levels} \\ \text{Highest scenario value} \end{array}$$

Dose Calculation Inputs

These parameters are used as inputs to the "ShieldDose2" model, but only when 'FluxType' = 1PtDiff (differential flux). At least one of the 'DoseRateOut' or 'DoseAccumOut' parameters is required to be specified. Dose calculations are not applicable for the 'Plasma' model and/or *uni-directional* flux results. More details about the ShieldDose2 model parameters may be found in [Seltzer, 1994]. The dose results produced are for the input flux particle species, either protons (trapped, *not* solar), or electrons (sum of electron and bremsstrahlung contributions).

Note that the ShieldDose2 "bremsstrahlung" data tables being used by the CmdLineAe9Ap9 application have been updated to correct an error present in the original SHIELDOSE2 publication. In the 'elbrbas2.dat' file, with the exception of the *Al* detector targets, the *finite slab* and *semi-infinite slab* data tables had been switched [Heynderickx, 2013].

Parameter Keyword Name	Allowed Values	Required for Dose calc.	Default Value	Description
DoseDepths	0.111 – 111.1 (mm) 4.374 – 4374 (mils) 0.03 – 30.0 (g/cm^2)	Required	None	Comma-separated list of Aluminum shielding thickness depths, in units specified by the DoseDepthUnits parameter. At least *three* depth values should be specified, and in increasing order, due to requirements of the ShieldDose2 model. They will be sorted automatically if not in the proper order. Depth values are recommended to be limited to these ranges specified; validity of results for depths outside these ranges is uncertain.
DoseDepthUnits *DoseDepthU* *(deprecated)*	millimeters | mm mils gpercm2	Optional	millimeters	Units of Aluminum shielding thickness depth
DoseGeometry *DoseDetGeom* *(deprecated)*	spherical finiteslab semiinfiniteslab	Optional	spherical	Geometry of Aluminum shielding in front of (or around) detector - ½ dose at center of solid aluminum sphere - dose at transmission surface of finite aluminum slab - dose in semi-infinite aluminum medium

Parameter Keyword Name	Allowed Values	Required for Dose calc.	Default Value	Description
DoseDetector *DoseDetType* *(deprecated)*	aluminum | Al graphite silicon | Si air bone calcium | Ca gallium | Ga lithium | Li glass | SiO2 tissue water | H2O	Optional	silicon	Detector material type
DoseAttnMode *DoseAttnMd* *(deprecated)*	none nuclearinteractions nuclearandneutrons	Optional	none	Nuclear attenuation mode for dose calculations - no nuclear attenuation - nuclear attenuation, but neglecting neutron energy transport - nuclear attenuation including neutron energy transport
ShieldDose2DB DoseModelDB	*<path>*†/sd2DB.h5 †<path> may be specified using an environment variable	Optional	*< path from MagFieldDB>/ sd2DB.h5*	Database file for the ShieldDose2 model; when not specified, it is assumed that this file is located in the same directory as the MagFieldDB parameter specification. This parameter is only needed when using a non-default IGRF or SD2 database files/locations, or other specialized model runs.

Orbit Propagation Inputs

When the appropriate set of these orbit parameters is specified, the generated ephemeris information is written to the file specified by the 'OrbitFile' parameter (in the *Basic Model Inputs* group), **overwriting the ephemeris file if it already exists.** The ephemeris time and position information (supplied or generated) is always included in all model output files.

See Appendix F for information about the orbital element parameters, and the content and usage of the Two-Line Element (TLE) files.

Parameter Keyword Name	Allowed Values	Usage Coordination	Default Value	Description
OrbTLE	Valid path† and file name of Two Line Elements (TLE) file †path may be specified using an environment variable	Conditionally Required*	None	Path and filename of the TLE file used to generate the ephemeris information. When using the 'Kepler' propagator, the TLE file may contain only one TLE set. See Appendix F for a description of Two-Line Element files. *Orbit propagation requires that either OrbTLE, or the OrbElemTime with its set of associated orbit element parameters, must be specified.
OrbStart	Time value, in one of the supported time specification forms*.	Always Required	None	Start date and time of ephemeris to be generated. *See the 'Supported Time Specifications' table on page 16. Values are comma-separated, when requiring multiple fields.
OrbEnd	Time value, in one of the supported time specification forms*.	Always Required	None	End date and time of ephemeris to be generated. *See the 'Supported Time Specifications' table on page 16. Values are comma-separated, when requiring multiple fields.
OrbStep	>0 seconds	Always Required	None	Time interval (in seconds) between ephemeris orbit positions. Recommended values, according to general orbit type: LEO: 10 sec MEO: 300 sec HEO: 60 sec GEO: 3600 sec.

Parameter Keyword Name	Allowed Values	Usage Coordination	Default Value	Description
OrbPropType	SGP4 SatEph Kepler	Always Required	None	Orbit propagator to be used for generating the ephemeris information. 'SGP4' is a commonly-used orbit propagator. 'SatEph' is another name for the Lokangle propagator, developed and used by AFRL for decades. 'Kepler' is a very basic orbit propagator with optional J2 perturbation effects. See the descriptions of these propagators on page 31.
OrbMode	Standard Improved	Optional [SGP4 only]	Standard	Run mode for 'SGP4' propagator.
OrbDatum	72Old 72 84	Optional [SGP4 only]	84	WGS Datum constants to be used by 'SGP4' propagator. [WGS = World Geodetic System spheroidal definition of the Earth.]
OrbUseJ2	True False	Optional [Kepler only]	True	Use J2 perturbations in 'Kepler' propagator calculations.
OrbElemTime *OrbElmTim* *(deprecated)*	Time value, in one of the supported time specification forms. [See the 'Supported Time Specifications' table on page 16. Values are comma-separated, when requiring multiple fields.]	Conditionally Required*	None	Date and time associated with the specified orbital element values. *Orbit propagation requires that either OrbElemTime (with a set of associated orbit definition parameters) or OrbTLE, must be specified.
The following parameters are used to define an orbit; specify all values for one of the following types: 'mean', 'solar', 'classical', 'state vector' or 'geosynch'				
OrbIncl OrbInclin	0 – 180 (degrees)	'mean' or 'solar' or 'classical'	0	Orbital inclination (degrees)

Parameter Keyword Name	Allowed Values	Usage Coordination	Default Value	Description
OrbRAAN OrbRtAsc *OrbRAAsNd* *(deprecated)*	0.0 – 360.0 (degrees)	'mean' or 'classical'	0	Right ascension of ascending node (degrees) Also known as the '*celestial* longitude of the ascending node'.
OrbEcc* OrbEccen	0.0 – 1.0	'mean', 'solar', or 'classical'	0	Eccentricity (unitless) *[OrbAltPer+OrbAltApog can be interchanged with OrbEccen+OrbSemiMaj, for 'solar' or 'classical' sets of elements]
OrbArgPerig OrbArgPer	0 – 360 (degrees)	'mean' or 'classical'	0	Argument of perigee (degrees)
OrbMeanAnom OrbMeanAn	0 – 360 (degrees)	'mean', 'solar', or 'classical'	0	Mean anomaly (degrees) (alternatively, the OrbTimePer parameter may specified)
OrbMeanMot OrbMeanMo	>0.0 – 30.0	'mean'	None	Orbital mean motion (revolutions/day)
Orb1stDerMM *Orb1stDer* *(deprecated)*	-10 – +10	'mean'	0	First derivative of mean motion (rev/day^2) [only used by SatEph propagator]
Orb2ndDerMM *Orb2ndDer* *(deprecated)*	-1 – +1	'mean'	0	Second derivative of mean motion (rev/day^3) [only used by SatEph propagator]
OrbBStar	-1 – +1	'mean'	0	Ballistic coefficient [only used by SGP4 propagator]
OrbAltPer*	>0 – ~10xRe	'solar' or 'classical'	None	Altitude of orbit perigee (km) *[OrbAltPer+OrbAltApog can be interchanged with OrbEccen+OrbSemiMaj]
OrbAltApog* OrbAltApo	>0 – ~10xRe	'solar'or 'classical'	None	Altitude of orbit apogee (km) *[OrbAltPer+OrbAltApog can be interchanged with OrbEccen+OrbSemiMaj]

Parameter Keyword Name	Allowed Values	Usage Coordination	Default Value	Description
OrbSemiMaj* *OrbSmjAxis* *(deprecated)*	>0 – ~10xRe	'solar' or 'classical'	None	Semi-major axis (Re) of orbit *[OrbAltPer+OrbAltApog can be interchanged with OrbEccen+OrbSemiMaj]
OrbLTApog *OrbLocTimeApo* *(deprecated)*	0 – <24 (hours)	'solar'	None	Local time of orbit apogee (hours.frac)
OrbLTMaxIncl *OrbLocTimeMaxIncl* *(deprecated)*	0 – <24 (hours)	'solar'	None	Local time of maximum inclination [ie, max latitude] (hours.frac)
OrbTimePer OrbTimPerig	Time value, in one of the supported time specification forms*.	*Optional for* 'mean', 'solar', or 'classical'	None	Time of orbit perigee, *used in place of* OrbMeanAnom. *--any value specified for OrbMeanAnom will be overridden--* *See the 'Supported Time Specifications' table on page 16. Values are comma-separated, when requiring multiple fields.
OrbPosXyz	|<64,000km|	'state vector'	None	Comma-separated position coordinate values (X,Y,Z in GEI [km]) at orbital element time.
OrbVelXyz	|<50km/sec|	'state vector'	None	Comma-separated velocity vector values (X,Y,Z in GEI [km/s]) at orbital element time.
OrbGeoLon	-180 – 360 (degrees)	'geosynch'	None	Geographic East longitude (degrees) at orbital element time.

Orbit Propagator Descriptions

- The ‘SatEph’ (or 'Lokangle') propagator has been developed and used by researchers at the US Air Force Research Laboratory (AFRL) for several decades. It accounts for secular and periodic perturbations, and gravitational effects. When using a TLE file, this propagator performs interpolation of the orbital elements between adjacent TLE entries. This propagator is able to compute ephemeris at times *prior* to the element time. For its orbit degradation calculations, the Mean Motion First and Second Derivatives are used; the Bstar value is ignored.
- The ‘SGP4’ (Simplified General Perturbations 4) propagator (sometimes called SPACETRACK) considers secular and periodic variations due to Earth oblateness, solar and lunar gravitational effects, gravitational resonance effects, and orbital decay using a drag model. When using a TLE file, the latest TLE entry for the given time is used, and may exhibit a slight discontinuity in the ephemeris position at the time of next TLE entry. For its orbit degradation calculations, the Bstar value is used; the Mean Motion First and Second Derivatives are ignored.
- The 'Kepler' propagator is very basic orbit propagator that applies no perturbations, except for 'J2' effects, if selected. The 'J2' perturbation accounts for secular variations in the orbit due to oblateness of the Earth. *Because of its minimal or no perturbation effects, the use of the Kepler propagator is strongly recommended for the generation of long-duration ephemeris information with stable orbit characteristics.* By neglecting higher-order physics, this propagator simulates the effects of *station-keeping* maneuvers. Support of TLE files is limited, as the file may contain only one entry.

Both the 'SGP4' and 'Kepler' propagators are 'forward-generating' only. That is, they are unable to compute ephemeris at times prior to the specified 'Element Time', or prior to the first time value in the supplied TLE file.
The application of the higher-order physics effects in the ‘‘SatEph’ and ‘SGP4’ propagators will cause the orbital characteristics of the generated ephemeris to gradually change when the propagation is performed for an extended duration beyond the ‘Element Time’ or the last time value in the supplied TLE file.

Orbit Ephemeris File Description

The ephemeris file, required for performing the model calculations, is generated by one of the available orbit propagators, or may be supplied by the user (but must be in a supported format). The generated (output) or user-supplied (input) ephemeris file name is specified with the required 'OrbitFile' parameter.

The input 'ephemeris' file may alternatively be used to define a ***grid*** of coordinate positions, for a fixed date and time.

The date/time and coordinate information of the *output* ephemeris file may be controlled using several optional parameters. The date/time form (default='MJD') is specified by the 'TimeSpec' parameter, and the position coordinate system (default='GEI') is specified by the 'CoordSys' parameter. Additional formatting options are available using the 'CoordUnits' (default='Re'), 'CoordOrder' (default='Std') and 'DataDelim' (default='comma') parameters. See the full description of these parameters in the 'Advanced Model Inputs' table, starting on page 18. All model output files include the ephemeris information, and are also written using these same formatting specifications.

For proper processing of the specified *input* ephemeris (or coordinate grid) text file, the similar 'InTimeSpec', 'InCoordSys', 'InCoordUnits', 'InCoordOrder' and 'InDataDelim' parameters may be used to describe the formatting of the date/time and coordinate information it contains. If not specified, these parameters will default to their corresponding 'output' parameter settings. See the full description of these parameters in the 'Advanced Model Inputs' table, starting on page 18. The input ephemeris file is expected to contain the one or more columns needed for the time specification, followed by three columns of position coordinate values, with the specified data delimiter character between each of the values; any additional values beyond those columns are ignored. The file may contain any number of header lines (comments), designed with a '#' in the first column.

It is perfectly valid to specify different settings between the 'input' and 'output' formatting parameters when an input ephemeris file is used. When an ephemeris file is being generated, any input formatting parameters specified will be ignored.

As the positions in the generated/supplied ephemeris or grid file, specified by 'OrbitFile', are processed by the selected radiation belt model, omni-directional flux values are computed by default. To compute uni-directional flux values instead, one or more directions may be specified using one of the 'PitchAngles', 'PitchFile' or 'DirFile' parameters (see 'Advanced Model Inputs'). Use the 'PitchAngles' parameter to compute uni-directional flux values with the same pitch angle(s) for all positions in the ephemeris file, specifying one or more angles (in degrees). For uni-directional flux values with different pitch angle(s) for each of positions in the ephemeris

file, use the 'PitchFile' parameter, specifying a file containing a set of pitch angles associated with each entry of the ephemeris file. Alternatively, the 'DirFile' parameter may be used for specifying a file containing a set of direction vectors for each entry of the ephemeris file. For proper interpretation, these direction vectors *must* use the same coordinate system as the positions of the ephemeris file (Cartesian coordinate systems only). The direction vectors may be full magnitude or unit vectors. The pitch angles associated with each of these direction vectors are calculated during the processing, and included in the model output files.

Important note: Dose calculations (using the ShieldDose2 model) *require* **omni-**directional differential flux values (1PtDiff). Specifications for computing uni-directional flux values and/or integral flux values along with any dose calculation specifications will be flagged as an error.

Processing of the Input Files

Ephemeris Generation Processing

An ephemeris file may be generated, independent of any radiation belt model calculations, using the CmdLineAe9Ap9 application. The parameters required to be present in the input file are: 'MagFieldDB', 'OrbitFile', 'OrbPropType', 'OrbStart', 'OrbEnd', 'OrbStep', then either 'OrbTle' or 'OrbElemTime' and its associated set of orbit element parameters. The output formatting parameters ('TimeSpec', 'CoordSys', 'CoordUnits', 'CoordOrder', and 'DataDelim') may also be specified, if desired.

Alternatively, these ephemeris generation parameters may be included in the input file for the radiation belt model calculation. However, if the ephemeris is to be used with more than one radiation belt model (such as AE9 for electrons, then AP9 for protons), this would be calculating the ephemeris twice. This may be a concern for cases of significant ephemeris generation tasks. It is suggested to either generate the ephemeris using a separate input file, or include the ephemeris generation with the first model run input file, then just specify the resulting ephemeris file (for the 'OrbitFile' parameter), and its input formatting specifications, in the subsequent model run input file(s).

Model Calculation Processing

Depending on the results desired, more than one model run may be required. For each model run, a separate input file is required, particularly since each model has a specific database and valid energy level range, and possibly model-dependent options.

Because of the multiple input files required, and the (possibly multiple) output files associated with each of these input files being produced, it is suggested that a file-naming scheme is used so not to inadvertently overwrite output files and/or confuse the relationship between input and output files. The naming scheme used for the GUI application is shown at the bottom of page 51. Using such naming scheme, a typical sequence of commands needed to generate the desired set of results might look like: (in this example, "Run directory"='Run', and "Run name"='Zeta')

```
> CmdLineAe9Ap9 -i Run/Zeta.Ephem.input.txt
> CmdLineAe9Ap9 -i Run/Zeta.AE9.input.txt
> CmdLineAe9Ap9 -i Run/Zeta.AP9.input.txt
```

However, depending on types of calculations requested in these input files, further commands may be required to fully complete their processing, employing the included post-processing applications, described in the next section.

The construction of these input files and their subsequent execution by the CmdLineAe9Ap9 application, and any relevant post-processing, is done automatically by the GUI application. At this time, a few of the 'advanced' and 'accumulation' options are not currently supported by the GUI. However, the GUI can still be used to generate a set of input files containing a majority of the desired parameters (but halt the execution), for subsequent editing to add any remaining options.

Post-Processing of the Output Files

For certain types of desired calculations, the results from more than one model may need to be combined. These post-processing applications are used for performing this step.

Total Dose Calculations

Because these radiation belt models calculate the fluxes for electron and proton particles separately, their dose results are likewise calculated separately. Normally, the 'Total Dose' is just the sum of their dose results. However, the calculation of the aggregation confidence level for 'total dose' results requires that each of the scenario 'total dose' results need to be computed *prior* to their aggregation. The '`TotalDose`' utility application is available to perform the summation of these particle-specific dose results, and includes the special handling of the aggregation dose results. This utility produces an additional set of dose output files, prepending 'Total' to the 'doserate' and/or 'doseaccum' descriptors in the output filenames. Following the example command sequence shown in the previous section, the next command would be:

```
> TotalDose Run Zeta
```

See the '`TotalDose_Readme.txt`' file in the executable directory for more information about the utility. This post-processing step is performed automatically within the GUI application when the model run includes dose calculations.

Tandem AE9/AP9 and Plasma Model Calculations

The 'AE9/AP9' model and 'Plasma' model (for the 'electron' and 'H+' species) may be used in tandem to provide results over a broader energy range. However, because these two models were developed independently, the results where their energy ranges overlap will not always match. When used together, it is recommend that the 'Plasma' model energy levels specified be less than 0.04 MeV for electrons, and less than 0.1 MeV for protons. *[Plasma model valid for $2 \le Lm \le 10$]*

The 'Plasma' model differential flux results may be used as-is. However, to obtain true ***integral*** flux results, information for the energy levels that are above the 'Plasma' model energy limits needs to be included. Pseudo-integral flux Plasma model runs may be performed at the desired energies, using 'FluxType'='2PtDiff', with the 'Energies2' list values all set to 0.04 MeV for electrons or 0.1 MeV for protons. Integral flux 'AE9/AP9' model runs are required at 0.04 MeV (and other energies, if desired) for electrons and 0.1 MeV (and others) for protons, using the same ephemeris information as the Plasma runs. When all runs have been executed, use the '`IntegralPlasma`' utility application to perform a post-processing adjustment of the Plasma results. This rewrites the Plasma integral output files, revising the '2PtDiff' values to be 'Integral' values, incorporating the respective 'AE9/AP9' integral results at their lowest energies. Following the example command sequence in the previous section, the next command would be:

```
> IntegralPlasma Run Zeta
```

More information about this utility application may be found in the '`IntPlasma_Readme.txt`' file in the executable directory. This post-processing adjustment operation is performed automatically within the GUI application for these types of model calculations.

Model Output File Description

The input file specifications define the types of model calculations to be performed, and the corresponding ASCII output files that are generated. The names of the output files indicate the type of data it contains, its calculation mode, and potentially the aggregation/division.

The names of these model output files are constructed from the required '<OutFile>' and '<FluxOut>' parameter values, other optional model output parameter values, and any model output accumulations and/or aggregation parameter specifications. The basic output file name assembly scheme is below. This scheme ensures unique output file names that also provide descriptive information about their contents.

Prefix	Data Mode based on <FluxOut> value	Data Type	Percentile, Scenario and/or Aggregation Id, based on <FluxOut> and <Aggregate> values	Suffix
<OutFile> (without the filename extension, ie '.txt')	_mean	Based on <*Out> and AccumMode values. *See the table below*	(-n/a- for mean)	filename extension of <OutFile> or '.txt' if not included
	_pctile		_## (percentile, in <FluxOut> value)	
	_pert _mc		_### (scenario identification #) *_agg_mean (usage is discouraged)* _conf_level_##	

AccumMode DataType	*Cumul*	*Interval*	*Full*	*Boxcar*	*Expon*
Flux	_flux	_fluxIntvAvg	_fluxFullAvg	_fluxRunAvg	_fluxExpAvg
Fluence	_fluence	_fluenceIntv	_fluenceFull	- - - - - - -	- - - - - - -
DoseRate	_doserate	_doserateIntvAvg	_doserateFullAvg	- - - - - - -	- - - - - - -
DoseAccum	_doseaccum	_doseaccumIntv	_doseaccumFull	- - - - - - -	- - - - - - -

The output file generated when using the advanced 'AdiabatOut' option is named in the form '<OutFile>_adiabat.txt'. See Appendix E for a description of its output values.

Each model output file contains several header lines (comments, defined by '#' in first column) that identify the model, its parameters, type of output values, and other pertinent information (but not necessarily the complete set of model parameters specified in the input file). The last header line specifies the data column labels and units. Each data line contains the date/time and coordinate information, followed by one or more data values, as appropriate for the file's data type/mode/aggregation. If uni-directional flux results are being calculated for more than one direction or pitch angle, multiple lines with the same time code will be present in the file, one for each direction or pitch angle.

Note: Any specified accumulation interval will affect the frequency of results in the fluence, dose rate and/or accumulated dose output files. The position information for these entries will be shown as *zero values*, since the associated data no longer corresponds to a single discrete point.

The date/time and coordinate information of the model output file (in tandem with the generated ephemeris file, if applicable) may be controlled using several optional parameters. The date/time form (default='MJD') is specified by the 'TimeSpec' parameter, and the position coordinate system (default='GEI') is specified by the 'CoordSys' parameter. Additional formatting options are available using the 'CoordUnits' (default='Re'), 'CoordOrder' (default='Std') and 'DataDelim' (default='comma') parameters. See the full description of these parameters in the 'Advanced Model Inputs' table, starting on page 18.

Any of the generated ASCII output files may be converted to a corresponding file in the Microsoft Excel format using the 'ConvertToXlsx' utility, included in the distribution. See the description for this application on page 39. This utility is also able to convert the generated ASCII ephemeris and adiabatic invariant output files, as well as the plot files produced by the GUI application.

The 'Advanced Inputs' section includes the optional parameters 'WorkDir', 'BinDirName' and 'DelBinDir'. These control the location and disposition of the intermediate binary data files that are used as part of the normal processing. By default, these binary files are removed at the completion of the calculation tasks. If these files are desired to be retained for subsequent use (outside of this software package), set the 'DelBinDir' parameter to 'false'. Unless a directory name is specified using 'BinDirName', these binary files will be located in a subdirectory named in the form of "`temp_`<OutFile *file name prefix*>`_p`<*process#*>", in the directory specified by 'WorkDir' or that in 'OutFile'. Please note that any post-processing that may be required for certain types of data calculations is only performed on the ASCII files that are derived from these binary files.

The names of these binary data files are simply the names of their corresponding ASCII files with a '`.bin`' suffix. Each binary data file has an associated ASCII file containing the file's header text, and has a '`.bin.hdr`' filename suffix. The binary data file format consists of records of 8-byte floating-point values that match the values from each line of its corresponding ASCII file, with the exception that the time is always stored as a Modified Julian Date value.

Processing Performance Tuning

When executing the CmdLineAe9Ap9 application on systems with limited amounts of free memory, the overall performance may be improved by specifying an additional parameter in the model run input file. The (optional) tuning parameter 'ChunkSize' (default value = 240) defines the number of orbital positions being processed during each call to the lower-level model routines. The size of this processing 'chunk' directly relates to the amount of internal memory needed, beyond the model overhead. Specifying a lower value, such as 120, should improve performance times on limited-memory systems. Lower values will also increase the frequency of progress updates. Moderately larger values may be used on systems with adequate resources.

Starting with the v1.35.001 release, the CmdLineAe9Ap9 application includes parallel-processing capabilities (except Win*32* platforms), employing the Message Passing Interface (MPI) protocols. The input file 'NumProc' parameter specifies the total number of processors to be used for the model run – this count *includes* one processor for the "master control" node. Therefore, a minimum value of 3 is needed for using two processors for the data calculations. The number of existing processors on the host system is queried through system calls, and the number to be used is capped at this limit, if exceeded. For *cluster environments*, this hardware query should be bypassed by specifying a negative number instead; using *more* processors than physically present will *degrade* the processing performance. Model runs on a cluster environment may also require that the processing nodes be 'reserved' or 'allocated' before they may be used; consult the cluster's administrator and/or documentation. Single-threaded model operation is performed if the 'NumProc' parameter is not present in the file, or specified as '1'.

Alternatively, the number of processors to be used for parallelized processing may be specified as a command-line argument. This value will *override* the 'NumProc' value specified in the input file. The same rules for 'NumProc' regarding the hardware query for processors present also apply to this command-line argument value.

During all model runs, the CmdLineAe9Ap9 application uses binary-format intermediary files, which are written and read from a temporary directory. By default this directory is located where the final ASCII-formatted output files are to be written. An alternate location for this temporary directory may be specified using the 'WorkDir' parameter, but must be read/write accessible by all processing nodes. This may improve model performance if this alternate location has faster read/write speeds than the default location, due to disk hardware capabilities and/or network configuration. If the model 'Run' and/or 'Work' directories are located on RAID-5 disk units, the use of the 'NumFileIo' parameter may also improve the performance of multi-threaded model processing operations.

Separate parallelized model runs may be executed simultaneously on the same machine or cluster, provided the appropriate number of processors and an adequate amount of memory are available to support them concurrently.

ConvertToXlsx Utility Application

The ConvertToXlsx.py utility application, located in each of the platform executable directories, is a Python script for converting any of the model run ASCII output files into their equivalent forms in a Microsoft Excel file format. The output files generated by the 'Plot' tab of the GUI application (described later in this document) are also supported by this script.

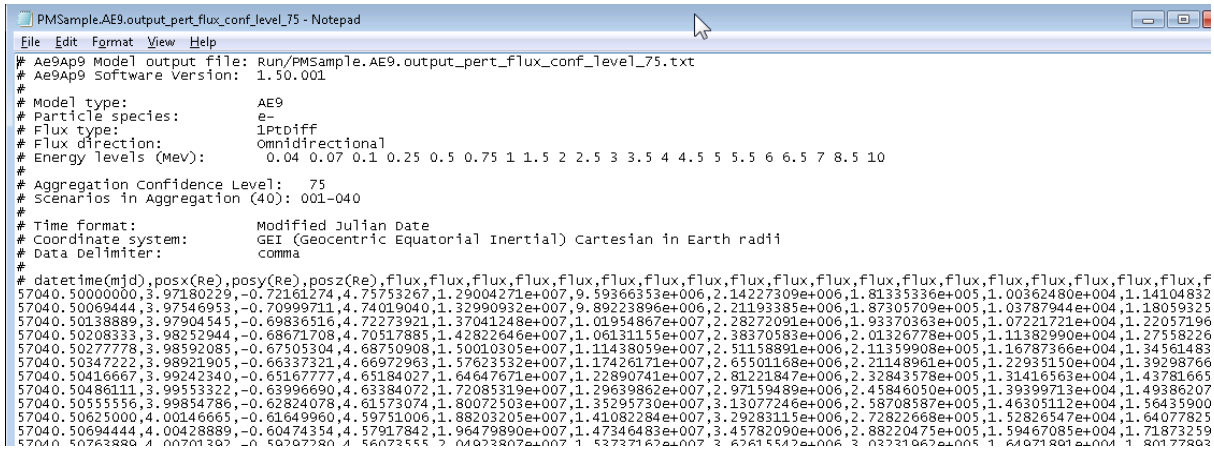
To utilize this conversion script, the host machine (Windows or Linux) must have the Python scripting language (v2.5 or later), with the 'XlsxWriter' Python module installed (see https://xlsxwriter.readthedocs.io/getting_started.html). Also see Appendix G.

The syntax for executing this script is simple: arguments are simply the name, or names, of the model run output files to be converted; these file names may include wildcards.

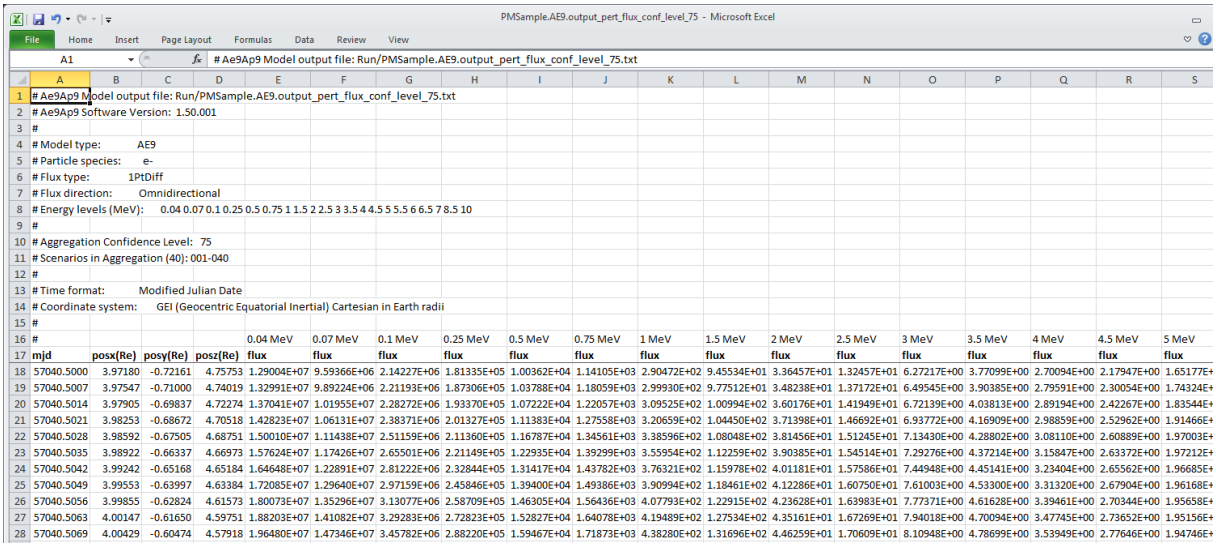
```
python ConvertToXlsx.py <filename1> [<filename2> … <filenameN>]
```

The resulting *.xlsx files are written in the same directory as their input files. The file header information is preserved in the newly generated Excel file, as seen below:

A sample ASCII format model run output file



The corresponding Excel format file



Graphical User Interface Application

The Ae9Ap9Gui application provides a simple graphical user interface front-end for performing model runs using the CmdLineAe9Ap9 application. Based on the selections and specifications made in the user interface, the appropriate model run input files with the corresponding parameter settings are generated and then executed. Basic 2-D plots of the calculated model results may be produced.

For each model run, the user specifies a '*Run Name*' for a filename prefix, and a '*Run Directory*' in which to write the various files produced. These default to '`Run1`' and '`Run`', respectively. The filename prefix is used for both the model run input and output files. The directory is usually relative to the application starting directory, but could be an absolute directory path, and/or include an environment variable.

The GUI application employs a user-specific configuration file for defining the model database files and for retaining the last-used state of the GUI items. When absent, this configuration file is automatically generated; it is located at '`%AppData%/Ae9Ap9/Gui_v1.50.ini`' on Windows systems, or '`$HOME/.config/Ae9Ap9/Gui_v1.50.ini`' on Linux systems.

The 'Config' menu button, at the top right corner of the GUI application window, allows the user to revise these configuration settings. All GUI items can be reset to their default setting, and the retention of the last-used state of GUI items may be disabled, if desired. The model database and directory specifications may be updated if needed; these can include environment variables. Any environment variables used must be defined *prior* to invoking the GUI application. It is highly recommended to only use the model database files that are included with the distribution.

The GUI application may be started from a directory other than its installation binary executable directory if a 'shortcut' (or 'symbolic link' on Linux) is used. The model database specifications will need to be revised, changing the default relative paths to be absolute. See Appendix G for more information. Unless an absolute 'run' directory is specified, the model run files will be generated relative to this starting directory.

The specifications from previously-generated model run input files may be used to initialize the GUI application state, specifying the run directory and name as arguments on the command line:

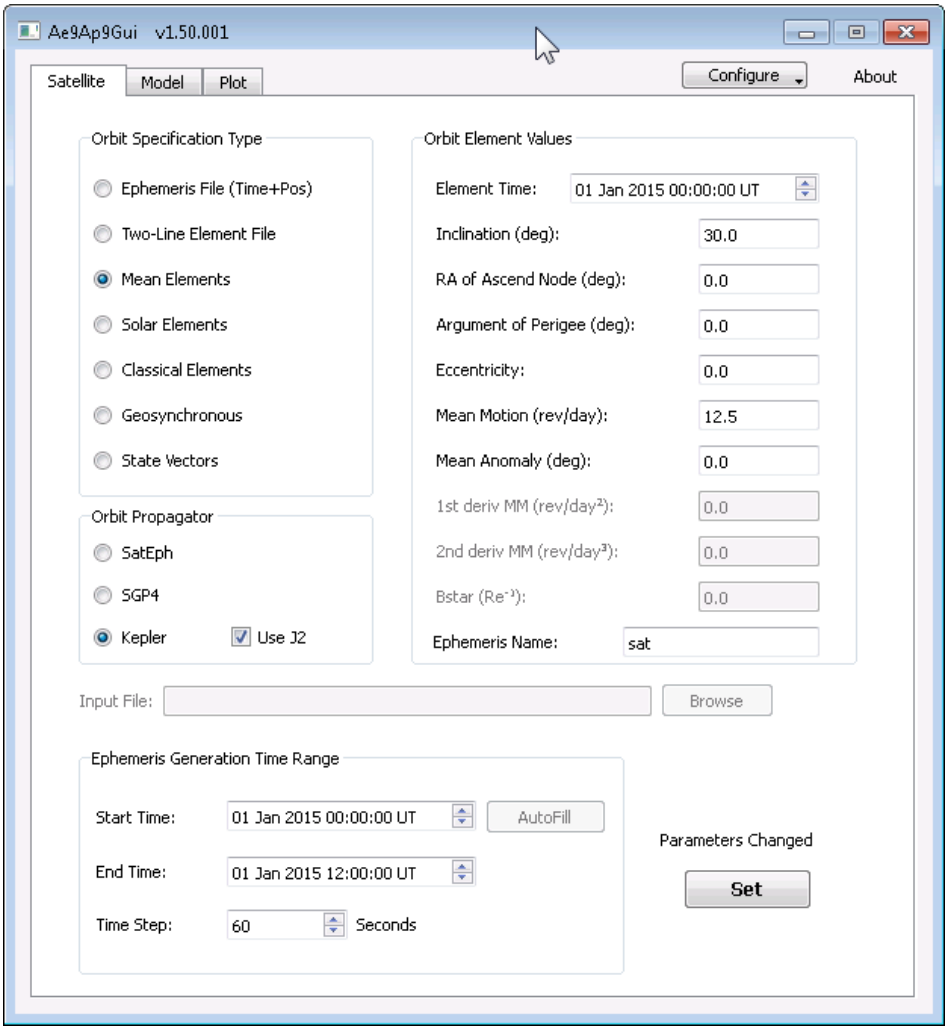
```
Ae9Ap9Gui [<RunDirectory> <RunName>]
```

Note: This feature does not support the use of pre-v1.35 model run input files, due to a change in the naming scheme of the input files and the numerous revised parameter keyword and value strings.

The GUI controls are divided into three tabbed pages, labeled '*Satellite*', '*Model*' and '*Plot*', following the typical progression of a model run session: an orbital path is defined on the 'Satellite' page; the radiation belt model is selected, configured and executed on the 'Model' page; the results of the model run are displayed using the 'Plot' page. The usage and available features on each of these pages are described in the following sections.

Satellite Tab

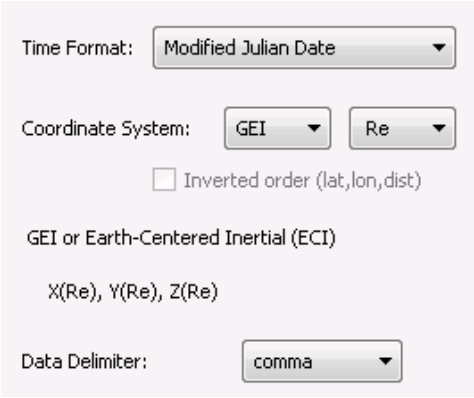
This page collects all necessary information for defining the times and orbital positions at which the radiation environment model values are to be calculated, usually along a satellite orbital path for a specific time period and increment.



The satellite orbital path may be specified in one of several different ways. Based on the '*Orbit Specification Type*' selected, the appropriate selections or value entry fields are shown on the right-hand column of the interface.

To use an existing ASCII-format ephemeris file, select '*Ephemeris (Time+Pos)*', and enter the filename in the '*Input File*' field. In order for the time and position information contained in this file to be correctly processed by the CmdLineAe9Ap9 application, its formatting must be described. Several different time forms, coordinate systems, and data delimiters are supported, and may be described through the user interface selections that are shown when this type is selected. Please note the date/time values are restricted to the 01 Jan 1950 – 31 Dec 2049 range, corresponding to the Modified Julian Date range of 33282.0 – 69806.9999.

Input ephemeris files with variable time steps are supported. If an 'Accumulation Interval' (on the Model tab) is used with the calculated flux values, this interval time must be larger than the largest ephemeris time step.



At left are the user interface controls for defining the time format, coordinate system and units, and data column delimiter character of the specified existing ephemeris file.

The supported time formats and coordinate systems are shown in the tables on page 16.
The supported data delimiter characters are 'comma', 'space' and 'tab'.

Model calculations at a ***grid*** of coordinate positions for a fixed date and time may be also performed using the '*Ephemeris (Time+Pos)*' input type. However, these results will not be able to be displayed using the '*Plot*' tab section of this GUI application, since its graphing operations require a time progression.

To use a TLE file, described in Appendix F, select '*Two-Line Element File*', and enter the filename in the '*Input File*' field. The TLE file may contain multiple entries (except when using the Kepler propagator), but for a single vehicle only.

All other orbit specification types require the user to enter values for a particular set of orbital element values, along with an associated '*Element Time*' value. The number of element values required depends on the type selected. Please note the units for each of the orbital element values, in particular for '*Mean Motion*' (# revolutions per day). The '*Ephemeris Name*' field is used as part of the filename of the generated ephemeris file.

The '*Mean Elements*' fields request the same set of orbital element values that are found in a typical TLE entry.

The '*Solar Elements*' fields request orbital element values that define the orbit with respect to the sun's position *at the specified element time*; this is *not* a sun-synchronous orbit.

The '*Classical Elements*' fields request the classical or Kepler orbital element values.

The '*Geosynchronous*' field only requests the desired geographic (east) longitude.

The '*State Vectors*' fields request the satellite position coordinate and velocity vector values at the element time. These values are required to be in the GEI coordinate system, in units of 'km' and 'km/sec', respectively.

Three orbit propagators are available:

- The ‘SatEph’ (or 'Lokangle') propagator has been developed and used by researchers at the US Air Force Research Laboratory (AFRL) for several decades. It accounts for secular and periodic perturbations, and gravitational effects. When using a TLE file, this propagator performs interpolation of the orbital elements between adjacent TLE entries. For its orbit degradation calculations, the Mean Motion First and Second Derivatives are used; the Bstar value is ignored.
- The ‘SGP4’ (Simplified General Perturbations 4) propagator (sometimes called SPACETRACK) considers secular and periodic variations due to Earth oblateness, solar and lunar gravitational effects, gravitational resonance effects, and orbital decay using a drag model. When using a TLE file, the latest TLE entry for the given time is used, and may exhibit a slight discontinuity in the ephemeris position at the time of next TLE entry. For its orbit degradation calculations, the Bstar value is used; the Mean Motion First and Second Derivatives are ignored.
- The 'Kepler' propagator is very basic orbit propagator that applies no perturbations, except for 'J2' effects, if selected. The 'J2' perturbation accounts for secular variations in the orbit due to oblateness of the Earth. *Because of its minimal or no perturbation effects, the Kepler propagator is strongly recommended for the generation of long-duration ephemeris information with stable orbit characteristics.* By neglecting higher-order physics, this propagator simulates the effects of *station-keeping* maneuvers. Support of TLE files is limited, as the file may contain only one entry.

Both the 'SGP4' and 'Kepler' propagators are 'forward-generating' only. That is, they are unable to compute ephemeris at times prior to the specified 'Element Time', or prior to the first time value in the supplied TLE file.

The application of the higher-order physics effects in the ‘SatEph’ and ‘SGP4’ propagators will cause the orbital characteristics of the generated ephemeris to gradually change when the propagation is performed for an extended duration beyond the ‘Element Time’ or the last time value in the supplied TLE file. In those cases where use of the Kepler propagator would be better suited for maintaining the orbital characteristics, an informational dialog will be shown.

The time limits of the ephemeris file to be generated are specified by the ‘*Start Time*’ and ‘*End Time*’ entries, with the ‘*Time Step*’ time increment. Please keep in mind that a small time step value will result in more ephemeris positions at which the model values are being calculated. Model runs for long time durations with small time steps could potentially require many hours, days or weeks to complete, depending on your computer system performance.

There are four common satellite orbit types: Low Earth Orbit (LEO), Medium Earth Orbit (MEO), Highly Elliptical Orbit (HEO), and Geosynchronous Earth Orbit (GEO). LEO satellites orbit the earth at an altitude between approximately 200 and 2000 km. MEO satellites orbit the earth at an altitude generally between 2000 and 20,000 km. HEO satellites orbit the Earth at altitudes which can vary between approximately 2000 and 39,000 km. GEO satellites orbit the Earth at a specific altitude of 35,786 km, completing an orbit in the same time as for the Earth to complete a full rotation.

Recommended time step sizes for adequate resolution in the model results along the orbital path:			
LEO 10 seconds	**MEO** 300 seconds	**HEO** 60 seconds	**GEO** 3600 seconds
These times are based on the variations of the calculated flux values that can be expected within the portions of the radiation belts that these orbits generally pass through.			

The optimal time step size will depend on exact characteristics of the established satellite orbit being used.

If the TLE file input type is selected, the '*Autofill*' button can be used to automatically fill the ephemeris generation time limits. When pressed, the contents of the specified TLE file are scanned. Assuming no file errors are detected, the '*Start Time*' is set to match to the first TLE entry time value, and the '*End Time*' is set to be the last TLE entry time value plus one day.

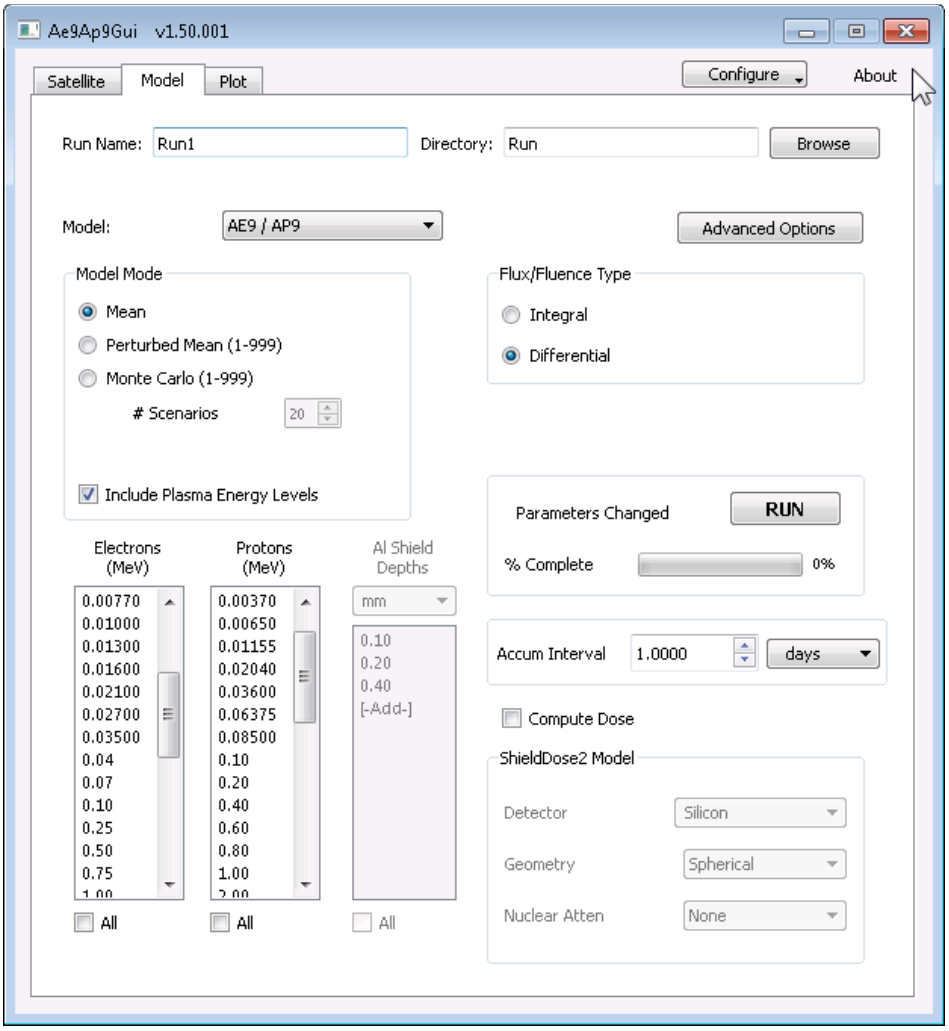
The ephemeris file will be generated during the execution of the model run from the specified TLE file or orbital element inputs, using the selected orbit propagator, for the specified start and stop times and time step size. When using a TLE file, the filename of the generated ephemeris file will be in the form: '`ephem_<TLE_input_file>.dat`'. For all other orbit specification types, the '*Ephemeris Name*' field is used as part of the filename of the generated ephemeris file, in the form: '`ephem_<EphemerisName>.dat`'.

The time and coordinate values of the generated ephemeris file (and also all model run output files) are written according to the specifications in the 'Advanced Options' dialog window displayed, via the button on the Model tab (described in the next section). The default output file formatting is comma-separated values of time (in Modified Julian Date form) and orbital position values in the GEI (ECI) coordinate system, in units of Earth-radii (1 Re = 6371.2 km).

When all orbit definition selections and specifications have been completed, press the '*Set*' button. The various inputs are checked, ensuring each of the entered orbital element values and time values are valid and properly coordinated. If applicable, the specified input file is also scanned, confirming its contents are in the expected or described format and that the values are within their respective acceptable ranges. Limited tests are also performed on the sets of orbit element values to determine if a sustainable orbit has been defined. If any type problem is detected, an informative message is shown in a dialog box. Depending on the selected propagator, general orbit type and ephemeris propagation duration, the use of the Kepler propagator may be recommended instead, if the conditions warrant.

Model Tab

This page collects all user-specified parameters required for calculating the various model values at the defined ephemeris positions. Each set of model run input and output files will be written in the specified '*Run Directory*' (relative to the GUI application starting directory, or absolute, and may include an environment variable), with their file names containing the prefix specified in the '*Run Name*' entry.



Five models are available from the '*Model*' drop-down box – 'AE9/AP9', 'Plasma (SPM)', and three "Legacy" models: 'AE8/AP8', 'CRRES ELE/PRO' and 'CAMMICE/MICS'. Depending on which model is selected, the appropriate parameter selections are shown in the GUI window.

An '*Accumulation Interval*' may be specified, in units of days or seconds. Its default is 1 day. This interval affects how often the fluence (and dose, if selected) values are calculated using the flux data values. When greater than zero, this accumulation interval must always be greater than the ephemeris time step. An accumulation interval of '-1' (days or seconds) indicates that the accumulation is over the entire time period, as defined by the ephemeris inputs.

For the 'AE9/AP9', 'AE8/AP8' and 'CRRES ELE/PRO' models, dose calculations may also be performed using the calculated particle flux and fluence results (*omnidirectional only* for 'AE9/AP9'). This feature is activated by checking the '*Compute Dose*' checkbox, which enables the selection of the "ShieldDose2" model parameters, and also automatically selects all energy levels of both electrons and protons. When using an '*Accumulation Interval*' value of "0.0", the dose calculations are performed at every time step, which may significantly lengthen the overall model execution time. An interval value of '0.25 days' (or '21600 seconds') would direct the dose and fluence calculations to be performed using the accumulation of flux values for the preceding 6 hours of ephemeris information. Additional information about the other "ShieldDose2" model parameters may be found in the 'Dose Calculation Inputs' section on page 25 of this document, and [Seltzer, 1994].

The values in the '*Shield Depths*' list may be displayed in units of '*g/cm*^{2}', '*mm*' or '*mils*'. Using the drop-down box to select a different unit will automatically convert the existing entry values to the new units. At least *three* depth values must be selected for proper calculation results.

The values comprising the '*Shield Depths*' list may also be customized. Double-click on an entry to edit its value (or alternatively, use Shift+Ctrl+click). When the editing of an entry is complete, its position in the list will automatically be adjusted to maintain increasing numerical order. Entering "0" or blank will delete the entry. New entries may be added to the list by selecting the special '*[-Add-]*' entry at the bottom of the list. The list's pop-up "tooltip" information shows the expected range of valid values, as well as these editing instructions. These customized list values may be saved to a file by pressing 'Shift' when clicking on a list entry. A saved list of values may be reloaded by pressing 'Ctrl' when clicking on a list entry. Only valid list entries from appropriate list files will be loaded. Appropriate error, warning or informational dialogs will be displayed as needed.

The 'AE9/AP9' and 'Plasma' models provide a choice of different modes of flux calculations to be performed. The 'Mean' (and 'Percentile', within the 'Advanced Options') modes capture the statistical behavior of the data upon which the model was built. The 'Perturbed Mean' mode adds the uncertainties in the mean flux maps that are due to measurement and gap-filling errors. The 'Monte Carlo' mode (not available for the 'Plasma' model) contains all these uncertainties of the 'Perturbed Mean' mode, then adds an estimate of the dynamic variations due to space weather processes. A "# Scenarios" is specified for these latter two modes, defining a set of "scenario" IDs (*1-N*, and are used as the random number seed), each producing a different flux profile, bounded by the variances due to measurement error and space weather. These variations represent the range associated with space weather on multiple timescales and span the variability observed throughout a solar cycle; however, the solar cycle phase is not reproduced. The use of these scenario "seed" numbers enable these model flux results to be fully reproducible, provided that the same ephemeris information is used. It is recommended that at least ten scenarios be used for producing statistically meaningful 'aggregation' confidence levels from these modes.

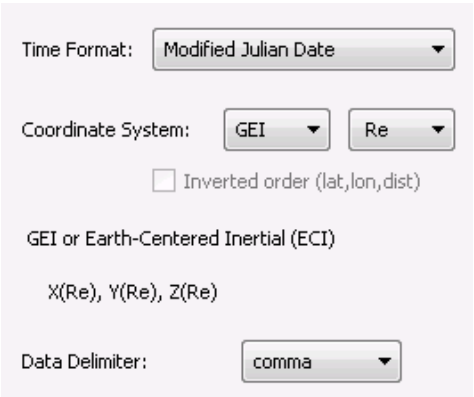
Tandem '**AE9/AP9**' and '**Plasma**' model operation:

When the '**AE9/AP9**' model is selected, the Electrons and Protons energy lists show two distinct sets of values within each list. Those energy values with less than 0.04 MeV (electrons) or 0.1 MeV (protons) [by default, these are shown with five digits to the right of the decimal point] are calculated using the SPM 'Plasma' model for 'electrons' and/or 'H+'(protons). The higher energy values [by default, shown with only two digits] are calculated with the AE9/AP9 model. "Monte Carlo"-type calculations will not be allowed if any of the 'Plasma' energy values are selected. These energies may be removed from the lists by toggling the provided checkbox. *[Plasma model valid for $2 \leq Lm \leq 10$]*

The calculation of *Integral* flux values of the 'plasma' energies will automatically invoke a calculation of the integral flux values of the AE9/AP9 at their respective lowest energy levels, if not already selected. The results produced for the plasma energy levels will be adjusted to incorporate the results of these lowest energy AE9/AP9 energies. This post-processing adjustment of the plasma results is performed automatically within the operation of the GUI. A progress bar in a dialog window is shown when there are many files to be processed. Manually-performed model runs of this type will need to use the '`IntegralPlasma`' utility for performing this post-processing adjustment. This utility verifies that all of the specific requirements of the associated model runs are met before the processing is performed.

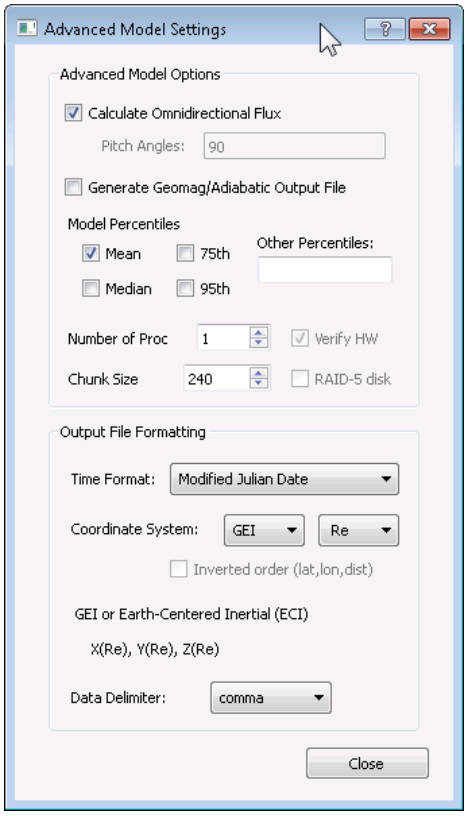
The energy lists (for the '**AE9/AP9**' and '**Plasma**' models only) may be customized in the same manner as that of the '*Shield Depths*' list. The energy list values used with the legacy models cannot be changed. The '**CAMMICE/MICS**' legacy model always calculates results for *all* of its pre-defined energy bins.

For all models, additional model run settings are available via the 'Advanced Options' button. When pressed, a dialog window is shown, containing two sections. The upper section deals with model-specific settings (if any), and the lower section permits the user to customize the formatting of the time and coordinate values that are written to the model run output files and the generated ephemeris file (if applicable). Press the 'Close' button when these settings are complete.



Shown at left is the user interface controls for defining the time format, coordinate system and units, and data column delimiter character for the model run output files and generated ephemeris file.

The supported time formats and coordinate systems are shown in the tables on page 16. The supported data delimiter characters are 'comma', 'space' and 'tab'.



The 'advanced' options for the 'AE9/AP9' or 'Plasma' models include the "*Calculate Omnidirectional Flux*" option, checked by default. When unchecked, the user may specify one or more local pitch angles at which the *unidirectional* flux values are calculated instead (please note that dose calculations require omnidirectional flux values). The "*Generate Geomag / Adiabatic Output File*" option, unchecked by default, is used to invoke the generation of a secondary ephemeris output file, containing geomagnetic parameter and adiabatic invariant coordinate values (MLT, B_{local}, $B_{equator}$, L_m, L*, Phi, K, H_{min}, $Alpha_{eq}$), as a function of local pitch angle. These values are calculated and used internally by the model routines; see Appendix D and E for more information. For omnidirectional model flux calculations, these values are reported for the 90° pitch angle only; for unidirectional, they are reported for each specified pitch angle.

The standard set of "*Model Percentiles*" or "*Confidence Levels*" (depending on the *Model Mode* selected) for the output files produced during the model execution is indicated by the state of the checkboxes. *Additional* percentiles or confidence levels may also be specified via the checkboxes and/or listing their values.

The "*Number of Proc*" specifies the number of processors to use (including the 'master' node) when performing multi-processor model calculations (*except Win32 platforms*). When the "*Verify HW*" checkbox is checked, the number of processors specified is compared to the actual number of processors present (as determined through system calls). When using multiple processors and the '*Run Directory*' (and/or '*Work Directory'*) is on a RAID-5 disk unit, checking the '*RAID-5 Disk*' checkbox improves the overall performance. The "*Chunk Size*" entry adjusts the tuning of the model run performance, particularly useful on computer systems with limited memory resources. See the 'Advanced Model Inputs' table on page 18 for more details.

The 'advanced' options for the 'AE8/AP8' or 'CRRES ELE/PRO' models include the "*Use Native Epoch*" option, checked by default, which imposes the use of model-specific year values (rather than the year specified by the satellite ephemeris) for the magnetic field model when performing the flux calculations. The "*Translate SAA*" option, also checked by default, is used to shift the South Atlantic Anomaly (SAA) feature from its 'native epoch' year location to that for the current year specified in the satellite ephemeris. See [Heynderickx et al, 1996] for more details on the effects of these settings.

There are no model-specific 'advanced' options for the 'CAMMICE' model.

To execute the model run, using the defined orbit, model parameters and settings, press the '*Run*' button. Before the execution begins, the various model parameter inputs are verified to contain proper and/or compatible settings. An informative error dialog is displayed if problems are detected. If the '*Run Name*' specification has been used before, a dialog box will ask if the previously-generated files may be overwritten. When the various selections have all been verified, a set of model run input files are generated, named in the form:
'*<RunDir>*/*<RunName>*.*<Model>*.input.txt'
where '*<Model>*' annotates the specific portion of the model (ie 'AE9' or 'AP9').
These files are used as input to the CmdLineAe9Ap9 application. Model runs that require multiple portions to be executed (ie, Electrons and Protons), are done in a serial manner (if specified, each of these model runs may be parallelized). When required, the generation of the ephemeris file will be performed prior to the first model run, and the resulting file is used as input for all other model runs. If an error occurs during one of the model run portions, any remaining portions will be canceled, and the model run files associated with '*Run Name*' will be marked as "incomplete". An error dialog will notify the user if this occurs. The 'Show Details' button in the error dialog permits the CmdLineAe9Ap9 application error messages to viewed, providing additional information regarding the failure conditions.

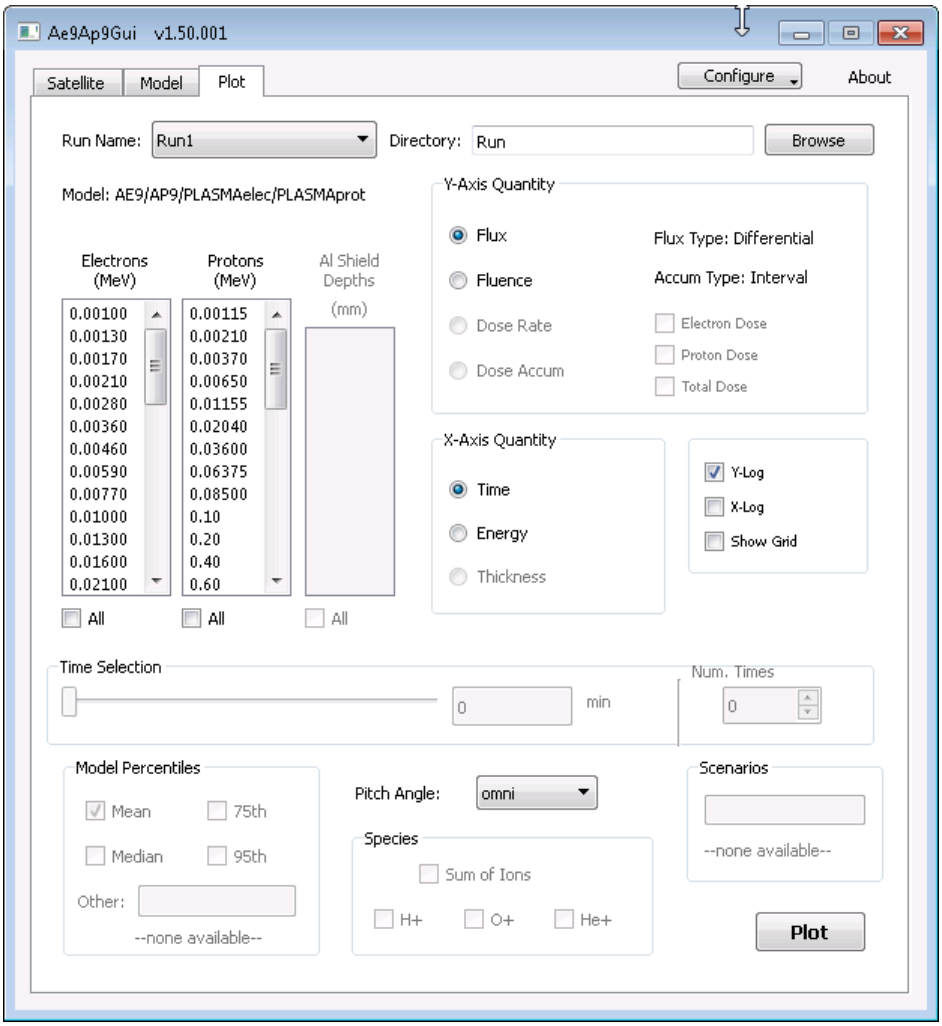
During the execution of the necessary model runs, the '*Run*' button will show '-*busy*-', and the run status will be shown in the '*% Complete*' progress bar. The update rate and frequency of this progress bar will vary, depending on the number of ephemeris positions being used, the model and species selected, and the types of calculations being performed. Adjustment of the model run performance tuning parameters, 'ChunkSize' and 'NumProc' (set in the 'Advanced Model Options dialog window), also affects this update frequency and execution speed. While "busy", the current satellite and model GUI selections may be viewed, but no changes are permitted. Depending on the model run parameters, a post-processing utility may be automatically invoked: for runs that include dose calculations, the TotalDose utility; runs that calculate the integral flux values at 'plasma' energies, the IntegralPlasma utility. At the successful completion of all required model runs, the button is changed back to showing '*Run*'.

The generated model output files are named in the form:
'*<RunDir>*/*<RunName>*.*<Model>*.output_*<type>*.txt'
where '*<Model>*' annotates the specific portion of the model (ie 'AE9' or 'AP9').
The various permutations for '*<type>*' depend on the model 'mode' settings, and are shown in the 'Model Output Files' section on page 36.

When the '*Geomag/Adiabatic Output File*' option has been selected in the model 'advanced' dialog window, only one 'Adiabat' file is generated. Its filename in the form '*<RunDir>*/*<RunName>*.*<Model>*.output_adiabat.txt'; however, its results are independent of the model '*<Model>*'; see Appendix E for a description of the parameters contained in the file.

Plot Tab

This page provides a method for producing basic 2-D plots of the model calculation results.



For the specified '*Directory*' location, all available '*Run Name*' model runs are shown in the drop-down list. These can be from this current or any previous GUI application session. Manually-configured and -executed model runs may also be selected, provided that the expected input and output file-naming form has been used:

The input file is named in the form: "`<RunName>.<Model>.input.txt`", and contains the 'OutFile' parameter value in the form "`<RunName>.<Model>.output.txt`".

Where `<Model>` is one of : 'AE9', 'AP9', 'PLASMA_E', 'PLASMA_O', 'PLASMA_H', or 'PLASMA_HE', and/or:

- for AE9/AP9/SPM "tandem" runs : 'PLASMAelec' and 'PLASMAprot'
- for legacy models: 'AE8', 'AP8', 'CRRESELE', 'CRRESPRO' or 'CAMMICE'.

Based on the input and output files of the selected '*Run Name*', the model name and flux type are identified, and the lists for energy levels and shield depths used are appropriately populated. Other pertinent information, such as percentiles, confidence levels, scenario collections, pitch angles and/or species are also shown. Additional details of the full set of model parameters are always available from the model run input files.

If the selected '*Run Name*' model run has been marked as "incomplete" (due to an error during its generation), a warning dialog is displayed. Attempts to plot from such a set of model run files may show incorrect results and/or cause application instability.

The selection methods for the desired energy levels or shield depth values will depend on the type of plot desired: versus '*Time*', '*Energy*' or '*Thickness*'. The model run parameters may also dictate additional selections that are needed, such as pitch angles, species, scenario numbers or percentiles/confidence levels.

For plots of values versus '*Energy*' or '*Thickness*', two time-slice specification methods are available. The '*Time Selection*' slider allows a specific time value (whole minutes from the starting time) within the dataset to be selected. Alternatively, when the '*Num Times*' spinbox is changed from zero, this specifies the number of evenly-spaced time slices to plot, the first one always at time "t=0" of the time period.

Press the '*Plot*' button when selections are complete. An informative error dialog is displayed if additional selections are required. The plot is displayed in a new window; the GUI will remain frozen until this plot window is closed. However, if the selected model results are all zeros, no plot will be produced and a notice explaining this is displayed instead. Based on the selections made, one or more sets of data values are plotted, each using a different color and/or dot/dash pattern. The key below identifies each of these lines. Fluence and dose values are plotted based on the '*Accumulation Interval*' specification, and so these graphs versus time will appear as steps rather than curves when a non-zero interval is used.

For each plot produced, an ASCII-formatted file of the data plotted is written in the same directory, using the same data delimiter used in the generation of the model output files. These plot files are suitable for use by other plotting applications. The file-naming form is: '`<RunName>_Plot_<###>.txt`', where '`###`' is simply the number of the plot generated *during the current GUI application session*. Parameter labels for each data column are included in these plot data files, but do not provide model run parameters. These details are always available in the associated '*Run Name*'-prefix model input files.

To use these output files in Excel, see the description of the '`ConvertToXlsx`' utility on page 39.

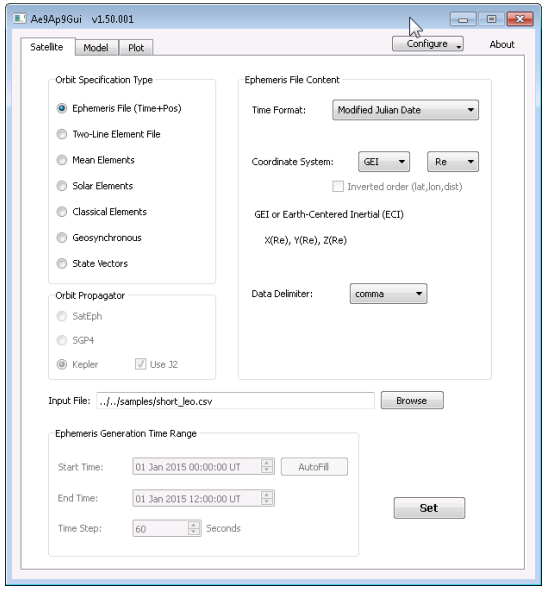
Example GUI-based Model Runs

Several examples of using the GUI to perform model runs and produce basic plots are shown in the series of screenshots on the following pages.

Example 1: MeanSample

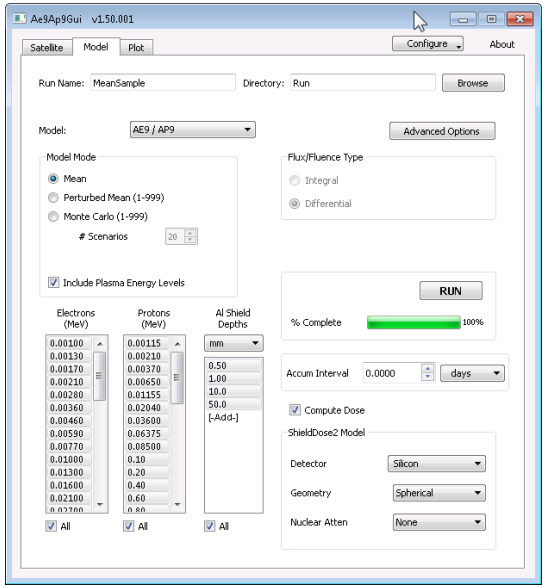
Satellite Tab:

-Select ‘Ephemeris File’ type
-Enter ‘../../samples/short_leo.csv’ for the Input File
-Set Time Format to ‘Modified Julian Date’, Coordinate System to ‘GEI’ and ‘km’, and Data Delimiter to ‘comma’
-Press the ‘Set’ button



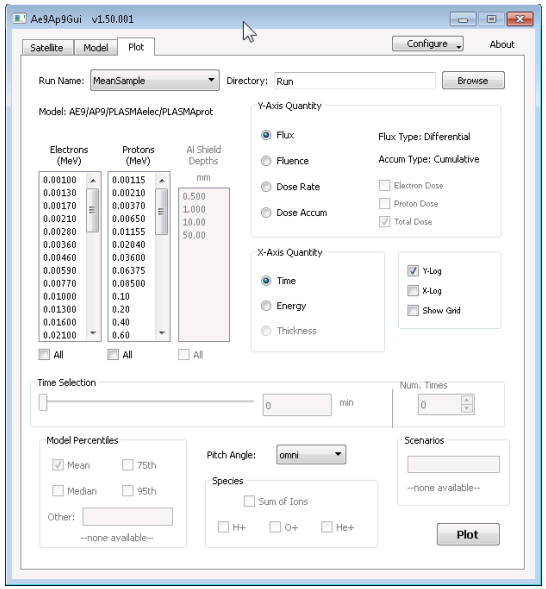
Model Tab:

-Enter ‘MeanSample’ for Run Name
-Specify a directory for the output files
-Check ‘Compute Dose’ checkbox
(all electron and proton energies will automatically be selected)
-Double-click on the first four lines in the ‘Shield Depths’ list, entering the values of 0.50, 1.0, 10.0 and 50.0 (units = ‘mm’)
-Check the ‘All’ checkbox below the Shield Depths list
-Set ‘Accum Interval’ to 0.00 days
-Press the ‘Run’ button
-The progress bar will show the run completion status.



Plot Tab:

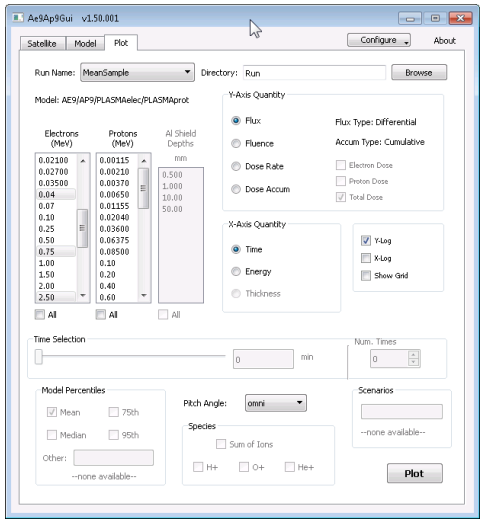
At the conclusion of the model run, switch to the ‘Plot’ tab; the Run Name will automatically be set to the ‘MeanSample’ name. The electron and proton energy levels and shield depths, as selected on the Model tab, will populate their respective lists on the Plot tab.



The results from any other model runs performed via the GUI may also be accessed from the ‘Run Name’ drop-down list.

MeanSample, Continued (Plot Tab)

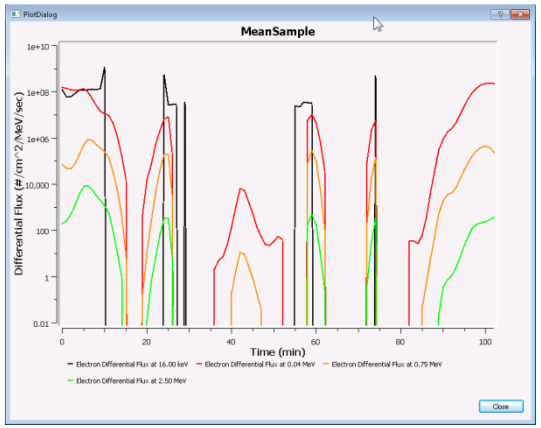
-Select a few electron energy levels
-Press the ‘Plot’ button



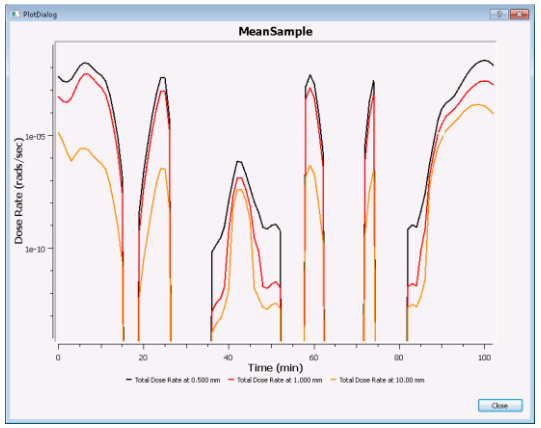
A pop-up window containing the graph will appear.

This user interface provides the ability to generate many different types of 2D plots from the available model run output data files, such as:

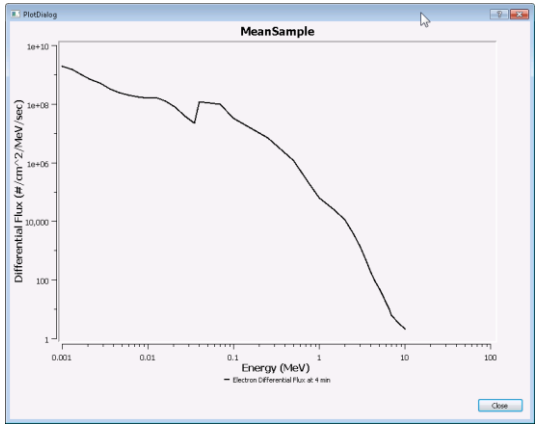
→Flux or Fluence vs Time



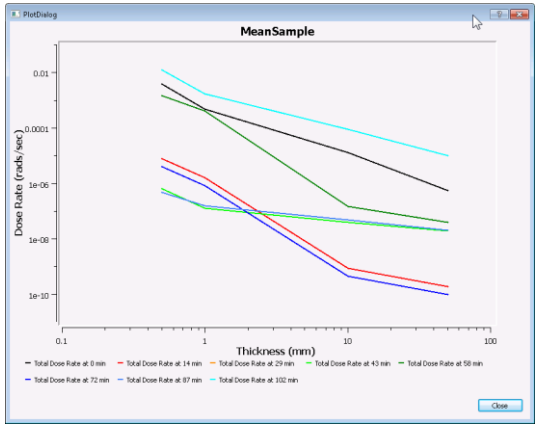
→Dose Rate or Dose Accum vs Time



→Flux or Fluence vs Energy,
at a single time, or multiple times



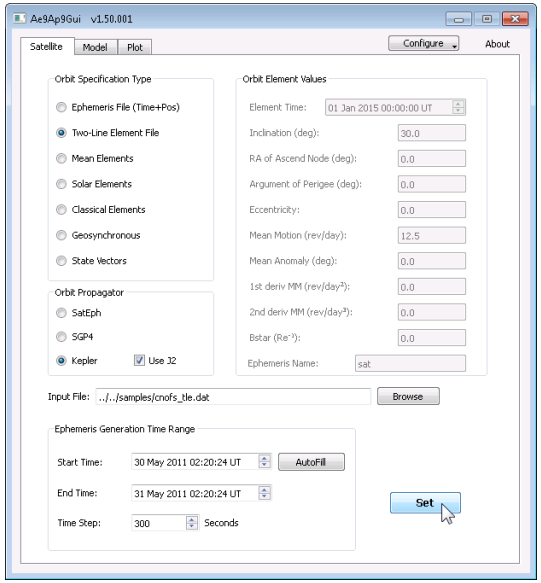
→Dose Rate or Dose Accum vs Thickness
at a single time, or multiple times



Example 2: MonteCarloSample

Satellite Tab:

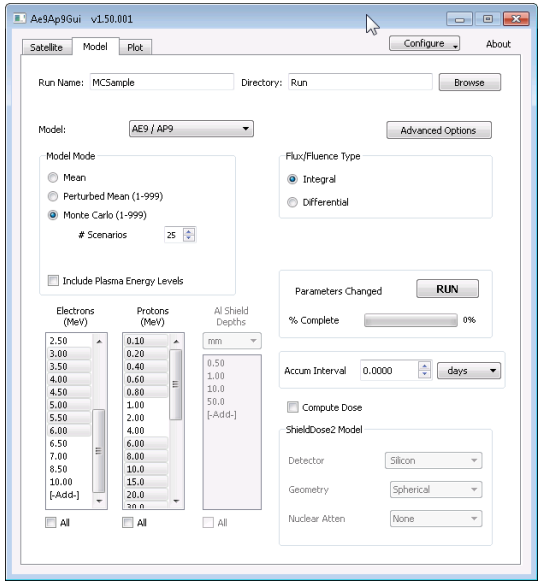
-Select ‘Two-Line Element File’ type
-Select ‘SatEph’ propagator
-Enter ‘../../samples/cnofs_tle.dat’ for the Input File
-Press the ‘Autofill’ button
-Set Time Step to be 300 seconds
-Press the ‘Set’ button



Model Tab:

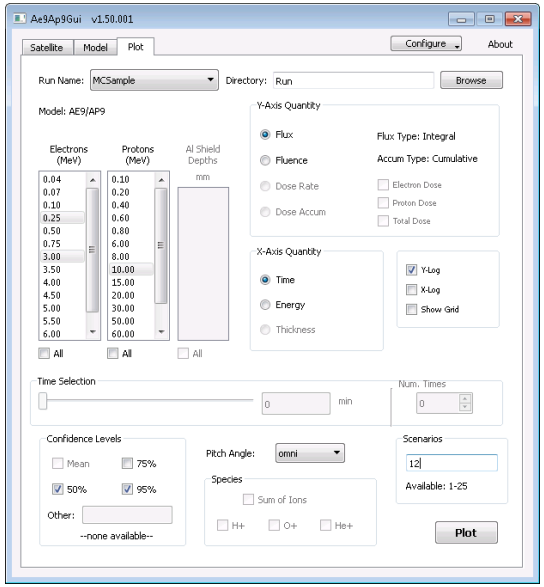
-Enter ‘MCSample’ for Run Name
-Select ‘Monte Carlo’ as Model Mode
-Set ‘# Scenarios’ to “25”

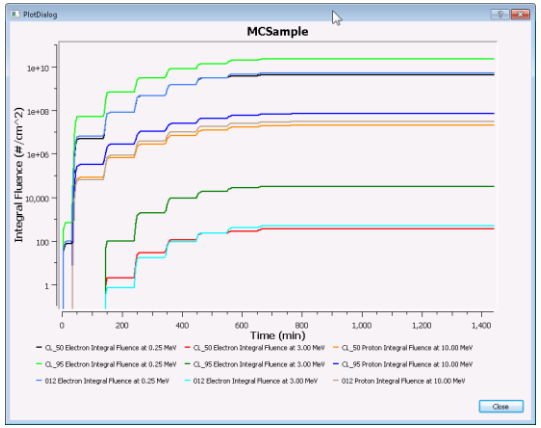
(uncheck ‘compute dose’ if previously checked)
-Select ‘Integral’ as Flux/Fluence Type
-Uncheck ‘Include Plasma Energy Levels’ checkbox
-Select electron energies 0.04 – 0.75 and 3.0 – 6.0 MeV
-Select proton energies 0.1 – 0.80 and 6.0 – 80.0 MeV
-Set ‘Accum Interval’ to 0.00 days
-Press ‘Run’, wait for run completion.



Plot Tab:

-Select ‘Fluence’
-Select Electron energies 0.25 and 3.0, and Proton energy 10.0 MeV
-Check ‘50%’ and ‘95%’ conf levels
-Enter ‘12’ in the Scenarios box
-Press the ‘Plot’ button

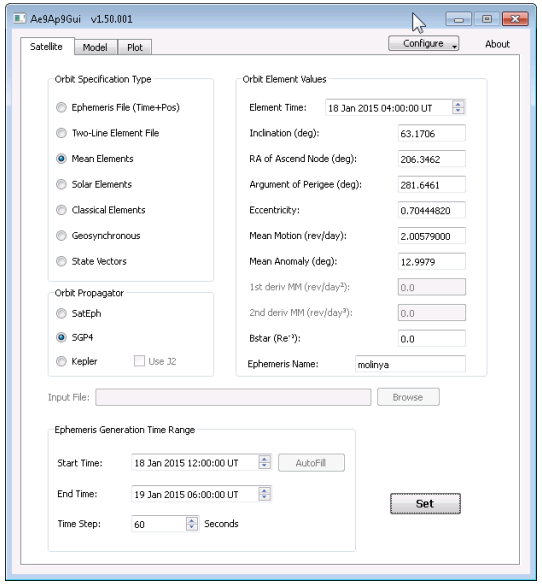




Example 3: PerturbedMeanSample

Satellite Tab:

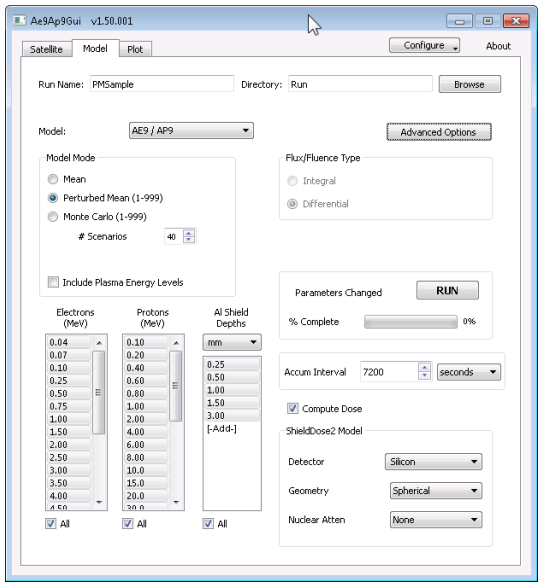
-Select ‘Mean Elements’ type
-Select ‘SGP4’ propagator
-Enter element values, time limits and time step value, as shown below
-Press the ‘Set’ button



Model Tab:

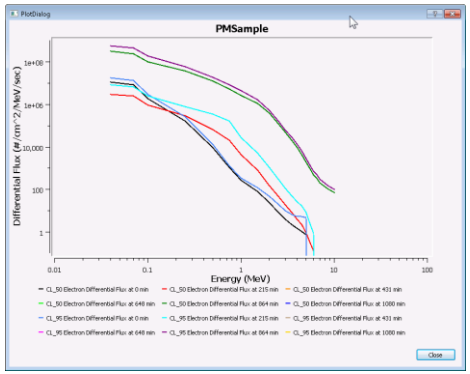
-Enter ‘PMSample’ for Run Name
-Select ‘Perturbed Mean’
-Set ‘# Scenarios’ to 40
-Uncheck ‘Include Plasma Energy Levels’

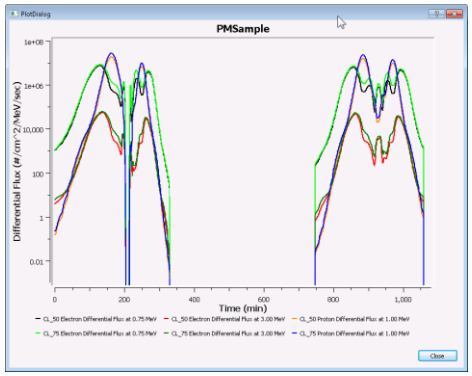
-Check the ‘Compute Dose’ checkbox
(all electron and proton energies will automatically be selected)
-Enter Shield Depths of 0.25, 0.50, 1.0, 1.50 and 3.00 mm
-Check 'All' under Shield Depths
-Enter 7200 *seconds* for the
‘Accumulation Interval’
-Press the ‘Run’ button

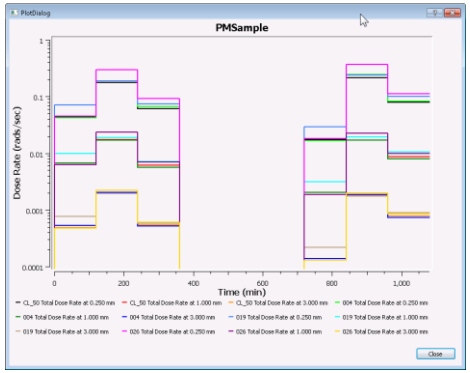


Plot Tab:

-Many types of plots are available
-Selections can be determined from each plot’s labels and legends







References:

Bhavnani, K.H., and R.P. Vancour, 11 Dec. 1991, "Coordinate Systems for Space and Geophysical Applications", Report PL-TR-91-2296, Phillips Laboratory, Hanscom AFB, Mass. [http://www.dtic.mil/dtic/tr/fulltext/u2/a247550.pdf].

Ginet, G.P., T.P. O'Brien, S.L. Huston, W.R. Johnston, T.B. Guild, R. Friedel, C.D. Lindstrom, C.J. Roth, P. Whelan, R.A. Quinn, D. Madden, S. Morley and Yi-Jiun Su, "AE9, AP9 and SPM: New Models for Specifying the Trapped Energetic Particle and Space Plasma Environment". Space Science Reviews, March 2013. [http://dx.doi.org/10.1007/s11214-013-9964-y]

Feynman, J., G. Spitale, J. Wang, and S. Gabriel, Interplanetary Proton Fluence Model: JPL 1991, J. Geophys. Res., 98, 13,281-13,294, 1993.

Rosenqvist, L., Hilgers, A., Evans, H., Daly, E., Hapgood, M., Stamper, R., Zwickl, R., Bourdarie, S., Boscher, D. "Toolkit for Updating Interplanetary Proton-Cumulated Fluence Models." J. Spacecraft Rockets, 42(6), 1077 - 1090, 2005.

M.A. Xapsos, P.M. O'Neill and T.P. O'Brien, "Near-Earth Space Radiation Models", IEEE Trans. Nucl. Sci., vol. 60, no. 3, pp.1691-1705, 2013. [http://dx.doi.org/10.1109/TNS.2012.2225846]

O'Brien, T. Paul, 25 March 2014, AE9/AP9 Guidance for Third-Party Developers, Aerospace Report TOR-2014-01204.

Seltzer, Stephen M. 1994, "Updated Calculations for Routine Space-Shielding Radiation Dose Estimates: SHIELDOSE-2," National Institute of Standards and Technology Publication NISTIR 5477.

Heynderickx, D., J. Lemaire, E.J. Daly and H.D.R. Evans, "Calculating Low-Altitude Trapped Particle Fluxes with the NASA Models AP-8 and AE-8", Radiation Measurements, Vol. 26, pp. 947-952, 1996.

Heynderickx, D., private communication, May 2013.

Vernov, S.N., E.V. Gorchakov, P.I. Shavrin, and K.N. Sharvina, "Radiation Belts in the Region of the South-Atlantic Magnetic Anomaly", Space Science Reviews, Vol. 7, No. 4, pp. 490-533, 1967. [http://dx.doi.org/10.1007/BF00182684].

Vallado, D.A., P. Crawford, R. Hujsak, and T.S. Kelso, "Revisiting Spacetrack Report #3: Rev 2", Reston, VA, American Institute of Aeronautics and Astronautics, 2006. AIAA 2006-6753-Rev2.

Appendix A: Legacy AE8/AP8 and CRRESELE/PRO Model Inputs

The parameters in this table define those needed for performing runs with these Legacy models. Additional parameters listed in the 'Orbit Propagation Inputs', 'Dose Calculation Inputs' and 'Accumulation Inputs' tables (except 'Aggregate') may be used with these models. The time, coordinate system and data delimiter specification parameters listed in the 'Advanced Model Inputs' table may also be used.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
ModelType	AE8 AP8 CRRESELE CRRESPRO	Required	None	Type of model to be run (requires corresponding database file specified in ModelDB parameter)
ModelDB	*<path>*†/radiationBeltDB.h5 †<path> may be specified using an environment variable	Required	None	Database file used to drive the model, including path. (all four legacy models use this same database file)
MagFieldDB	*<path>*†/igrfDB.h5	Required	None	Database file for the magnetic field model, including path
OutFile	valid path and file name prefix	Required	None	A path and filename "prefix" that will be used when generating the model output files. See the 'Basic Model Inputs' table for more details.
OrbitFile	valid path and file name of ephemeris file	Required	None	File containing time and position information (ephemeris). See the 'Basic Model Inputs' table for more details.
FluxType	1PtDiff Integral	Required	None	Type of flux values to be computed.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
FluxOut	True	Optional	True	Flux data to be generated / output.
FluenceOut	True False	Optional	False	Fluence data to be generated / output.
DoseRateOut	True False	Optional	False	DoseRate data to be generated / output.
DoseAccumOut	True False	Optional	False	DoseAccum data to be generated / output.
OutData (deprecated)	*Flux, Fluence, DoseRate, CumDose*	*optional*	*None*	*Specifies type(s) of data to be generated / output. This parameter is **deprecated**, superseded by the four preceding parameters*
Energies	AE8: 0.04 – 8.0 AP8: 0.1 – 250.0 CRRESELE: 0.65, 0.95, 1.60, 2.00, 2.35, 2.75, 3.15, 3.75, 4.55, 5.75 CRRESPRO: 1.5, 2.1, 2.5, 2.9, 3.6, 4.3, 5.7, 6.8, 8.5, 9.7, 10.7, 13.2, 16.9, 19.4, 26.3, 30.9, 36.3, 41.1, 47.0, 55.0, 65.7, 81.3	Required	None	Comma-separated list of energy levels, in MeV, at which flux values are to be computed, at each time step. Energy values for AE8 or AP8 are restricted to their model-specific ranges. For the CRRES models, only these *specific* energy levels (or a subset) may be used.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
LegActLevel *REActLvl (deprecated)*	AE8 or AP8: min, max CRRESPRO: active, quiet	Required*	None	Activity Level (*except for CRRESELE)
LegActRange† *REActRange (deprecated)*	5-7.5, 7.5-10, 10-15, 15-20, 20-25, >25, avg, max, *or* all	Crresele*	None	Activity Level for CRRESELE *this or 'Leg15DayAP'parameter is required
Leg15DayAp† *RE15DayAP (deprecated)* †Only one of these activity specifications may be used.	0.0 – 400.0	Crresele*	None	15 day average AP index for CRRESELE *this or 'LegActRange' parameter is required
LegFixEpoch *REFixEpoch (deprecated)*	True False	Optional	True	Use the model-specific fixed epoch (year) values for the magnetic field model in the flux calculations. It is *highly recommended* to use the default 'true' value. Un-physical flux results may be produced (especially at low altitudes) if set to 'false'. See [Heynderickx et al, 1996], for more information.
LegShiftSAA *REShiftSAA (deprecated)*	True False	Optional	True	Shift the SAA from its fixed-epoch location to the location for the current year of the ephemeris. See [Heynderickx et al, 1996], for more information. This option is ignored if LegFixEpoch is 'false'.

Appendix B: Legacy CAMMICE/MICS Model Inputs

The parameters in this table define those needed for performing runs with the CAMMICE/MICS model. This model is set to produce flux values for twelve pre-defined energy bins (1.0-1.3, 1.8-2.4, 3.2-4.2, 5.6-7.4, 9.9-13.2, 17.5-23.3, 30.9-41.1, 54.7-72.8, 80.3-89.7, 100.1-111.7, 124.7-139.1, 155.3-193.4 keV). Dose calculations are *not* available for this model. Additional parameters listed in the 'Orbit Propagation Inputs' and 'Accumulation Inputs' tables (except 'Aggregate') may be used with this model. The time, coordinate system and data delimiter specification parameters listed in the 'Advanced Model Inputs' table may also be used.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
ModelType	CAMMICE	Required	None	Type of model to be run (requires corresponding database file specified in ModelDB parameter)
ModelDB	*<path>*†/cammiceDB.h5 †<path> may be specified using an environment variable	Required	None	Database file used to drive the model, corresponding to the selected ModelType. Must include path to file (absolute, or relative to the CmdLineAe9Ap9 execution location).
MagFieldDB	*<path>*†/igrfDB.h5	Required	None	Database file for the magnetic field model, including path
OutFile	valid path† and file name prefix	Required	None	A path and filename "prefix" that will be used when generating the model output files. See the 'Basic Model Inputs' table for more details.
OrbitFile	Valid path† and file name of ephemeris file	Required	None	File containing time and position information (ephemeris). See the 'Basic Model Inputs' table for more details.
FluxOut	True	Optional	True	Flux data to be generated / output (this is always 'on').
FluenceOut	True False	Optional	False	Fluence data to be generated / output.

Parameter Keyword Name	Allowed Values	Required	Default Value	Description
CamMFModel *CIModel (deprecated)*	Igrf igrfop	Required	None	Magnetic field model to use with CAMMICE (igrfop = IGRF w/ Olson Pfitzer external field model)
CamDstData *CIDstData (deprecated)*	all filtered	Required	None	CAMMICE data filter: use all data, or data for DST > -100
CamSpecies *CISpecies (deprecated)*	h+, he+, he+2, o+, h, he, o, ions multiple comma-separated species may be listed on same line	Required	None	CAMMICE species for which to return flux data. This parameter may appear multiple times.
CamPAngle *CIPAngle (deprecated)*	0-10, 10-20, 20-30, 30-40, 40-50, 50-60, 60-70, 70-80, 80-90, 90-100, 100-110, 110-120, 120-130, 130-140, 140-150, 150-160, 160-170, 170-180, *or* omni	Required	None	Bin of pitch angles in degrees for which to return flux data in the CAMMICE model. (omni=omnidirectional)

Appendix C: Modified Julian Date

The Modified Julian Date (MJD) is an astronomical time convention that has the great advantage of being a continuous time variable, without the discontinuities introduced by the usual civil time convention of years, month, days, hours, minutes and seconds. This makes it ideal for computer manipulation of long time series.

The Modified Julian Date is derived from a much older system called 'Julian Date', which was defined as the time, in days, since *noon* (1200GMT) on 1 January 4713 BC. The MJD simply subtracts 2400000.5 from the Julian Date (the extra 0.5 shifts the start of days from middle to the beginning). Thus, the use of MJD requires only use of 5 (rather than 7) digits to the left of the decimal point. This offset makes the Modified Julian Date to be defined as the number of days since 17 Nov 1858, 0000GMT.

For example:

01 Jan 2000, 1200GMT is Julian Date 2451545.0, and Modified Julian Date 51544.5
10 Oct 2012, 0000GMT is Julian Date 2456210.5, and Modified Julian Date 56210.0
01 Jan 1950, 0000GMT is Julian Date 2433282.5, and Modified Julian Date 33282.0

Many tools and algorithms exists to convert between calendar date and time to Julian or Modified Julian Dates

http://www.onlineconversion.com/julian_date.htm
http://scienceworld.wolfram.com/astronomy/ModifiedJulianDate.html
http://www.csgnetwork.com/julianmodifdateconv.html

In Excel, if you have a date/time in cell A1, then the following formula will convert it to MJD (but you'll need to set the formula's cell format to "number")"

```
=A1-date(1950,1,1)+33282
```

This works because Excel uses a date serial that is a decimal number of days since some reference epoch.

Matlab also uses some reference epoch. This snippet of Matlab code will convert a date string to MJD:

```
mjd = datenum(date_string)+33282-datenum(1950,1,1);
```

The CmdLineAe9Ap9 application is limited to Modified Julian Dates in the range of 33282.0 – 69806.999 (01 Jan 1950 – 31 Dec 2049).

An important warning for users of SPENVIS – a *NON-STANDARD* definition for 'Modified Julian Date' is used:
"Finally, note that the Modified Julian Date (MJD) used in SPENVIS is defined as the number of days from 1st January 1950 00:00 UT."
see http://www.spenvis.oma.be/help/models/sapre.html

Appendix D: AE9/AP9/SPM Model Coordinates

Magnetic Field Model

The basis for all AE9/AP9/SPM model coordinate and mapping calculations is the International Geophysical Reference Field (IGRF) model for the 'main' magnetic field, in conjunction with the Olson-Pfitzer Quiet (OPQ77) model for the 'external' magnetic field, evaluated at the specified epoch date. The IGRF tables are updated at five-year intervals; the recent update with 2015 values and its associated derivatives are 'valid' only until the beginning of 2020. Therefore, any IGRF field model calculation results (and hence any model calculation results) for epoch dates beyond 01 Jan 2020 are *fixed to this date*. Investigations are also underway for determining if an alternate form of extrapolation of these tables beyond the five-year limit can be used to produce appropriate and realistic magnetic field model results.

The Olson-Pfitzer Quiet (OPQ77) external magnetic field model represents all major magnetospheric current systems, is valid for all tilt angles (i.e., angles of incidence of the solar wind on the dipole axis), accurately represents the total magnetospheric magnetic field for conditions of low magnetic activity, and covers to the magnetopause, up to a maximum geocentric distance of 15 Re.

Model Reference Grid

The empirical basis of the AE9/AP9/SPM models is a set of flux maps derived from data measured by particle detectors and dosimeters on board numerous satellites that have traversed, or continue to traverse, the radiation belts. These measurements, once properly calibrated and cross-correlated, were mapped to a reference grid. For the AE9/AP9/SPM model, the primary reference grid coordinates are:

- E – particle energy,
- K – modified second adiabatic invariant, capturing the particle's bounce motion,
- Φ – third adiabatic invariant, capturing the particle's drift motion,

The (K,Φ) coordinates are used to accurately map the variations in particle distributions across the Earth's magnetic epochs. The particle energy E was chosen (instead of the first adiabatic invariant μ) since most detector channels measure a range of both energy and local pitch angle α, making the spread of measured μ much wider than just the spread in E.

The more intuitive Roederer L-shell L^* may be calculated from Φ at a given magnetic epoch according using the equation:

$L^* = 2\pi k_0/\Phi$ where k_0 is the time-dependent magnetic dipole parameter. For this model implementation, the k_0 value is fixed (k_0=0.30119615 G-Re^3), corresponding to the date 01 Jan 2000, 0000GMT.

Although these adiabatic invariant values may be treated something like spatial coordinates, it should be noted that they are properties of the ***particles***. Different measurements taken at the same point in space may be associated with several different adiabatic invariant values, depending on the particles' energies and/or pitch angles, as well as the current magnetospheric conditions.

It has been determined that the (E,K,Φ) coordinate system does a poor job representing flux variations in and near the loss cone region. At low altitudes, the particle flux is controlled more by the thermospheric neutral density than by the magnetic field. Another complication is the difference between the bounce loss cone and the drift loss cone. To address these issues in the AE9/AP9/SPM model, the H_{min} parameter is used to map those regions below 1000 km altitude. 'H_{min}' is defined as the minimum altitude above the Earth's surface a particle reaches during its drift-bounce orbit [Vernov et al, 1967]. This parameter is obtained as a by-product of the Φ computation, and is much better than Φ for tracking variations in the particle distributions at low altitudes, where the flux gradients are large. By design, the (K,Φ) and (K,H_{min}) coordinate grids overlap, enabling mostly smooth transitions between them.

Adiabatic coordinates are also less useful for the mapping of lower-energy plasma fluxes, where there is a strong magnetic local time (MLT) dependence and effects from electric fields and plasma waves. Consequently, the more traditional McIlwain L-shell L_m and equatorial pitch angle α_{eq} is used for the space plasma models (SPM) instead of (K,Φ). Though MLT variations are substantial for energies less than approximately 100 keV, the main purpose of the SPM is to establish the statistics of particle flux exposure for satellites in orbit. Because the satellite ephemeris position timing, and hence MLT, is not always known beforehand, the development of SPM deemed the MLT dependence a secondary priority; future versions of SPM may eventually include this.

Calculation of the drift shells needed to evaluate Φ and H_{min} is computationally expensive, due to the requirement to perform an integration over an entire drift shell. Direct computation using the IRBEM-LIB routines was sufficient for construction of the flux maps since there are a relatively small number of satellite ephemerides to compute. However, in the general model application, where a user might evaluate many orbits for long periods of time, the computational load to convert those ephemerides to their flux map coordinates can quickly become unrealistic. A neural network interpolation algorithm has been developed to produce Φ and H_{min} without needing the drift shell integration; these values have an error of 1% or less. The only inputs required for this neural net algorithm are the date/time, satellite position and detector look direction (or pitch angle); from these, the I ($= K/\sqrt{B}$) and B_{mirror} (both are pitch angle-dependent) parameters are determined from a relatively inexpensive magnetic field line trace. The neural network is an integral part of the AE9/AP9/SPM model software, allowing Φ and H_{min} to be calculated almost as quickly as L_m.

Model Coverage Limits

The AE9, AP9 and SPM models will calculate flux values for given time, position and energy levels, provided they remain within their respective limits, as summarized in the following table:

Model Energy and Spatial Coverage Limits

Model	**AE9**	**AP9**	**SPM e^-**	**SPM H^+, He^+, O^+**
Energy Range	40 keV – 10 MeV	100 keV – 2 GeV	1 – 40 keV	1.15 – 164 keV
Range in L	$0.98 < L^* < 12.4$	$0.98 < L^* < 12.4$	$2 < L_m < 10$	$2 < L_m < 10$

While any energy level desired within the limits of the model may be specified, please note that these model databases were constructed using fixed set of energy levels. The calculation of flux values for the requested energy levels involves a linear interpolation between flux values at these fixed energy levels. These levels are shown in the following table:

Model	*Fixed Energy Levels*
AE9	0.04 0.07 0.1 0.25 0.5 0.75 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 8.5 10.0 MeV
AP9	0.1 0.2 0.4 0.6 0.8 1.0 2.0 4.0 6.0 8.0 10 15 20 30 50 60 80 100 150 200 300 400 700 1200 2000 MeV
SPM e^-	1.0 1.3 1.7 2.1 2.8 3.6 4.6 5.9 7.7 10.0 13.0 16.0 21.0 27.0 35.0 40.0 keV
SPM H^+, He^+, O^+	1.15 2.1 3.7 6.5 11.55 20.4 36.0 63.75 85.0 105.9 131.9 164.3 keV

More details about the construction and algorithms used in these models may be found in [Ginet, et al, 2013] ('*AE9AP9SPM_SSR*_Overview' in the 'documents' directory), and the VDL website (see page 2).

Appendix E: Geomagnetic / Adiabatic Invariant Parameter Output

The magnetic field model and adiabatic invariant values are normally calculated and used internally by the AE9/AP9/SPM model, but are generally not seen by the end-user. However, these values are accessible via 'AdiabatOut' parameter of the CmdLineAe9Ap9 application, the 'Generate Geomag/Adiabatic Output' checkbox in the Advanced Options panel of the Ae9Ap9Gui application, and/or the various 'getAdiabaticCoords' or 'ComputeCoordinateSet' methods described in the "AE9/AP9/SPM Model Application Programming Interface" (API) document.

Internally, during model calculations, only one of the three sets of model grid coordinates [(L_m,α_{eq}), (K, Φ) or (K, H_{min})] are determined, dictated by the model type, pitch angle and actual spatial location, with some model-specific limitations imposed. These methods described above provide the model-independent values for all three grid coordinate sets. Not all of these parameters are valid at *all* spatial locations; since '0' is a valid value for some of the parameters, the special value of '`-1e31`' is used to indicate an 'undefined' parameter value. The parameters B_{local}, B_{equat} and MLT are independent of pitch angle.

The parameter values produced by these methods are as follows:

- PitchAngle (degrees) – local pitch angle, either supplied as input or derived from direction vectors supplied as input; when omnidirectional flux calculations are being performed, this local pitch angle is set to 90°.
- B_{local} [nT] – local magnetic field strength at position
- B_{equat} [nT] – associated magnetic field strength at magnetic equator (also known as B_{min} or B_0)
- MLT [hours] – local time at the magnetic equator (B_{equat}) location
- L_m – McIlwain's L-shell value
- K [$\sqrt{G}$ Re] – second adiabatic invariant
- Phi (Φ) [G Re^2] – third adiabatic invariant
- L* – Roeder's L-shell value; calculated from Phi, as previously described
- H_{min} [km] – minimum altitude above the Earth's surface a particle reaches during its drift-bounce orbit
- $Alpha_{eq}$ (α_{eq}) [degrees] – equatorial pitch angle

Keep in mind that these magnetic field model and adiabatic invariant values produced here are based on the IGRF internal field plus the OPQ77 external field, as used in model development and operation. They do *not* represent results driven by dynamically varying solar wind and geomagnetic activity inputs.

Appendix F: Two-Line Element (TLE) Files

Two-Line Element (TLE) is a standard NORAD data format used to convey sets of orbital element values that describe the orbital motion of Earth-orbiting satellites. Current and archived TLE data for many satellites may be obtained from various online sources, such as http://www.celestrak.com and http://www.heavens-above.com .

Please note that the proper columnar location of values within the TLE sets is *essential*. An extra or missing space, or a stray 'tab' character, will shift their positions and cause these values to be incorrectly parsed.

NORAD Two-Line Element Set Format

TLE Line 1		TLE Line 2	
Column	Description	Column	Description
01	Line Number of Element Data	01	Line Number of Element Data
03-07	Satellite Number	03-07	Satellite Number
08	Classification (U=Unclassified)		
10-11	International Designator (Last two digits of launch year)	09-16	Inclination [Degrees]
12-14	International Designator (Launch number of the year)		
15-17	International Designator (Piece of the launch)		
19-20	Epoch Year (Last two digits of year)	18-25	Right Ascension of the Ascending Node [Degrees]
21-32	Epoch (Day of the year and fractional portion of the day)	27-33	Eccentricity (decimal point assumed)
34-43	First Time Derivative of the Mean Motion divided by 2	35-42	Argument of Perigee [Degrees]
45-52	Second Time Derivative of Mean Motion divided by 6 (decimal point assumed)	44-51	Mean Anomaly [Degrees]
54-61	BSTAR drag term (decimal point assumed)	53-63	Mean Motion [Revs per day]
63	Ephemeris type		
65-68	Element number	64-68	Revolution number at epoch [Revs]
69	Checksum (Modulo 10)	69	Checksum (Modulo 10)

Example Two-Line Element set (the identification number '32765' corresponds to the C/NOFS satellite):

```
1 32765U 08017A   11150.09749074 +.00010799 +00000-0 +47888-3 0 0797
2 32765 013.0015 105.8044 0295409 031.4522 330.3172 14.8643027916917
```

Understanding Orbital Elements

Below are some potentially helpful online resources for understanding the orbital element definitions:

http://en.wikipedia.org/wiki/Orbital_elements
http://www.braeunig.us/space/orbmech.htm
http://www.amsat.org/amsat-new/tools/keps_detail.php
http://marine.rutgers.edu/mrs/education/class/paul/orbits.html

TLE Usage by Orbit Propagators

The 'SGP4' and 'SatEph' orbit propagators may use TLE files that contain multiple (chronologically ordered) entries for a single satellite. The 'Kepler' propagator is currently restricted to TLE files containing only a *single* entry.
The 'SatEph' propagator performs interpolation between adjacent TLE entries for smooth ephemeris results. An ephemeris discontinuity corresponding to an orbital maneuver may be approximated with the insertion of an intervening 'thrust card': a TLE "line 1" with a 'T' in column 8 and the time associated with the maneuver.

When using the 'SGP4' and 'Kepler' propagators, the ephemeris generation start time must be greater or equal to the time of the initial TLE entry in the file.

Appendix G: Installation Tips

Environment Variables

The CmdLineAe9Ap9 and Ae9Ap9Gui applications, as well as their various 'helper' applications and the post-processing utilities, support the use of environment variables. Environment variables can be used to define disk and directory locations that might be different between computers, enabling the same script or input files to be used on multiple systems. They may also be used simply to shorten a very long directory path specification.

The environment variables may be referenced in the model run input files or GUI configuration file using the Windows (ie `%DATADIR%`) or Linux (ie `$DATADIR`) form. Any environment variables to be used must be defined *before* starting the application.

Installation Customization

The model software suite files are extracted from the distribution zipfile in a specific directory tree structure. This setup should be kept intact, as many of the included scripts depend on the data files being in a specific relative location to the application executable files.

If a common install location is to be used, accessible by multiple users, it is recommended that the entire directory tree is moved or copied to that location. Within the individual user's desired directory, make links ('shortcuts' on Windows or 'symbolic links' on Linux) to the common installation's CmdLineAe9Ap9 and Ae9Ap9Gui application executable files. Their execution through these links will automatically invoke the various 'helper' applications in the common install location as required. If the utility applications (IntegralPlasma, TotalDose and ConvertToXlsx.py) are to be invoked *directly*, make similar links for these also. Any model run input files must specify the proper location of the database files; this is a case where the use of an environment variable could be helpful. The distribution's 'sample' and 'unitTest' model run input files will need to be updated if they are to be run from the individual user's directory. The GUI application's configuration dialog for the model databases includes a button that will convert their default *relative* paths into their new 'absolute' paths.

Windows Python Installation Instructions

In order to use the ConvertToXlsx.py utility application, the Python scripting language is required to be installed on the machine. Depending on the install mode selected (individual user vs all users), Administrator privileges may be required.

For 64-bit Windows, download and install using:
https://www.python.org/ftp/python/2.7.13/python-2.7.13.amd64.msi .

For 32-bit Windows, download and install using:
https://www.python.org/ftp/python/2.7.13/python-2.7.13.msi .

➔ In the 'Customize Python' step of the installation, be sure to activate the "Add python.exe to Path" option (at the bottom of the list). This enables 'python' to be referenced just by its name, instead of requiring its full installation path to be specified every time.

The ConvertToXlsx.py script also requires the installation of two Python modules, 'numpy' and 'xlsxwriter'. Open a Command Prompt terminal window, then enter these commands:

```
C:\> pip install numpy
```
--install messages--
```
C:\> pip install xlsxwriter
```
--install messages--

Appendix H: Troubleshooting

Windows Execution Issues

Security Pop-ups

The first time the IRENE (Ae9Ap9) software is executed on a Windows machine, a pop-up from the Windows OS and/or the installed security suite, anti-virus or firewall program may appear, asking if a certain program is 'safe' or if it should be allowed to be executed. This query may be triggered because the software package launches 'helper' applications from within the primary Ae9Ap9Gui and CmdLineAe9Ap9 applications. The helper applications listed here should be permitted to execute:

AggregTask.exe	ConvertTask.exe	FluxTask.exe
AggregTaskMpi.exe	ConvertTaskMpi.exe	FluxTaskMpi.exe
CmdLineAe9Ap9.exe	EphemTask.exe	IntegralPlasma.exe
CmdLineAe9Ap9Mpi.exe	EphemTaskMpi.exe	LegFluxTask.exe
ConcatTask.exe	FlueDoseTask.exe	LegFluxTaskMpi.exe
ConcatTaskMpi.exe	FlueDoseTaskMpi.exe	TotalDose.exe

The user query pop-up from the security software could potentially disrupt the internal communications between these helper applications, and therefore cause the model calculations to 'hang'. If this occurs, the model run will need to be manually terminated, using the Windows Task Manager, and then restarted.

No Model Results Generated

Some users have reported their model runs using the Ae9Ap9Gui application appear to complete very quickly, but produce no results, and no error messages are shown. Most likely, this is due to a missing library on the Windows machine.

To confirm this condition, open a Windows Command Prompt window, and perform the steps described under the 'Installation Testing' section, on page 9 of the *User's Guide* document. If the error messages indicate the "`MSVCP110.dll`" library is missing from your computer, then install the "Visual C++ Redistributable for Visual Studio 2012" package, available on the Microsoft website:

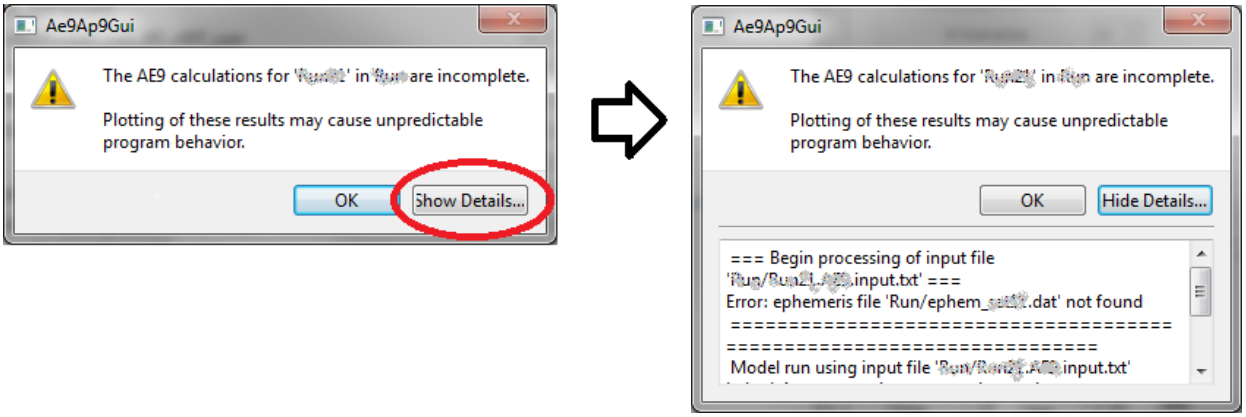
https://www.microsoft.com/en-us/download/details.aspx?id=30679

When/if the choice is presented, select "VSU_4\vcredist_x64.exe" for win64 machines, or "VSU_4\vcredist_x86.exe" for win32 machines.
This installation will require Administrator privileges.

Windows or Linux Execution Issues

Model Run Failures

For most common types of model run failures, informative messages are shown in the terminal window when using the CmdLineAe9Ap9 application. It is hoped that these messages provide an adequate description of the issue in order to correct the problem. When using the Ae9Ap9Gui application, these same informative messages are available by pressing the '*Show Details…*' button on the error dialog that appears:



Obtaining Help from the Model Team

When unable to determine the cause of the problem and/or the messages do not provide enough information to resolve the issue, contact the IRENE (AE9/AP9/SPM) model development team at ae9ap9@vdl.afrl.af.mil

When requesting assistance, please specify the version of the software package being used, the machine OS (Windows [7,8 or 10], [32 or 64-bit], or Linux [distribution name and version]), all error messages (from the terminal window or the GUI error dialog window), and at least a general description of the issue encountered. If relevant, also include the model run input files and/or samples of the output files produced.

Other Issues

Cluster-based Model Execution Tips

A *negative* number must be used when specifying the number of processors in the CmdLineAe9Ap9 arguments for a cluster-based execution; the negative number bypasses the *local host* hardware query; ie: `CmdLineAe9Ap9 -n -48 -i Run/BigJob.AE9.input.txt`

When using a Windows-based cluster, an additional argument may be needed to indicate the method of the MPI communication between compute nodes: specify '`-b`' to use SSH (recommended); otherwise, specify '`-c`' for using the Intel MPI '`hydra_service`' utility.
See https://software.intel.com/en-us/node/528873 for more information about this utility.
Use of this utility will require Administrator privileges.

If the allocated processors are 'tightly bound' to the executing threads, it may be necessary to increase the number of allocated processors by one, since the CmdLineAe9Ap9 application spawns the multi-threaded CmdLineAe9Ap9Mpi application, which then spawns off the additional various 'helper' applications.
Alternatively, this 'bootstrapping' step can be circumvented by directly invoking the CmdLineAe9Ap9Mpi application:

```
mpirun -np 1 CmdLineAe9Ap9Mpi -n -48 -i Run/BigJob.AE9.input.txt
```

The '`-np 1`' argument is required, so to invoke only *one* instance of the CmdLineAe9Ap9Mpi application. If using a Windows-based cluster, replace '`mpirun`' with '`mpiexec`'; if needed, also add the '`-b`' or '`-c`' MPI communications mode argument.

Linux Compilation Errors

Things to check when compilation errors occur:

- That all required third-party libraries are installed. Please compare the version numbers of these against those recommended in the *Build Instructions* document. In some cases, newer versions of these may not always be compatible with the Ae9Ap9 software.
- That the intended version of these libraries are being used (if more than one are present), and are properly specified in the respective '`internal_utils.cmake`' and '`ae9ap9gui.pro`' build configuration files, as described in the *Build Instructions* document.

To contact the IRENE (AE9/AP9/SPM) model development team, email ae9ap9@vdl.afrl.af.mil .

The IRENE (AE9/AP9/SPM) model package and related information can be obtained from AFRL's Virtual Distributed Laboratory (VDL) website: https://www.vdl.afrl.af.mil/programs/ae9ap9