
AE9/AP9/SPM

Radiation

Environment

Model

Release Notes

Version 1.35.001

Approved for public release; distribution is unlimited.

The 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.

AE9/AP9/SPM development team: Wm. Robert Johnston¹ (PI), T. Paul O'Brien² (PI), Gregory Ginnet³ (PI), Michael Starks¹, Stuart Huston⁴, Tim Guild², Christopher Roth⁴, Paul Whelan⁴, Rick Quinn⁴, Reiner Friedel⁵, Chad Lindstrom¹, Yi-Jiun Su¹, Steve Morley⁵, and Dan Madden⁶.

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

The 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

In a future release of AE9/AP9/SPM, the model will be renamed to be
“International Radiation Environment Near Earth” (IRENE).

Document Author: Christopher Roth/AER

Source code copyright 2016 Atmospheric and Environmental Research, Inc. (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

AE9/AP9/SPM Radiation Environment Model

Release Notes

Version 1.35.001

January 03, 2017

Highlights

Please refer to the 'Ae9Ap9_v1_30_001_ReleaseNotes' documents for a full description of the revisions and enhancements of the model since v1.20.004.

The internal architecture of the model processing software was significantly reworked to support parallelized processing of the model calculations. Multi-threaded operation is supported for Linux and 64-bit Windows platforms; however, it is *not* supported for 32-bit Windows platforms.

The CmdLineAe9Ap9 application parameter keyword/value inputs were revised to simplify their specification. Several new parameter options for the processing of model values were added. The naming scheme for the generated output files was revised to accommodate the new features.

A summary of the input parameter and file-naming changes is included in this document.

Two levels of API are now available: 1) application level, replicating nearly all options of the CmdLineAe9Ap9 program, including parallelized processing; 2) model level, permitting lower-level access to each of the calculation components: ephemeris, flux, fluence, dose and aggregation.

Software Changes

Significant internal architecture changes were implemented in the model software and its interfaces, largely in support of the implementation of parallel processing capabilities, and providing lower-level access to component model routines. The API was revised and expanded.

On the surface, the CmdLineAe9Ap9 program behavior is mostly unchanged. In previous versions, this was a large, monolithic application that performed all aspects of the processing to produce the requested model outputs. In this new architecture, the CmdLineAe9Ap9 application is a small program that directs the execution of a collection of other applications, each of which perform a specific set of calculation tasks, to produce the requested model outputs. This new data processing method, described in a later section in this document, occurs in both the single- and multi-threaded execution modes.

Several new keyword/value pairs were added to the parameter list for the CmdLineAe9Ap9 application for the specifications for new data processing features and capabilities. Many existing keywords now accept multiple values, and/or ranges of values, simplifying the construction of input files. Some parameter keyword and/or value strings were changed for consistency, readability, clarity and/or merged to reduce the complexity of the input files. The older keyword/values are still accepted for

backward compatibility, but their continued use is discouraged. The use of environment variables is now supported for the specification of filenames and paths. To properly distinguish the data results calculated using the new features, the naming scheme for the generated output files was revised.

The aggregation calculation of ‘percentile’ values has been relabeled as ‘confidence levels’. The allowable percent range has been expanded from 1-99 to 0-100, where the values reported for the 0 and 100 endpoints are the minimum and maximum values, respectively, of the aggregation scenario output values. The actual range of ‘confidence levels’ is determined by the number of scenarios.

The previous requirement that the ephemeris information be at fixed time steps was eliminated. This enables the use of different time step sizes, depending on the segment and/or altitude of the vehicle trajectory. However, large time gaps within the ephemeris will affect any accumulation calculations.

The handling of the orbital element mean motion first and second time derivatives was adjusted to apply the proper scaling. These values are only used by the SatEph propagator, and not by SGP4.

The ‘accumulation time’ interval specification now applies to the fluence calculation, as well as the dose values. The accumulation calculations perform interpolation of flux values at the beginning and end of each accumulation interval, as required. The accumulation interval duration may now be specified in units of days or seconds.

Two new flux accumulation modes are now available in the CmdLineAe9Ap9 application, but are currently deemed *experimental*: ‘Boxcar’ provides a windowed running average flux over an accumulation interval; ‘Exponential Average’ provides the exponential average flux that is an infinite response moving average filter.

The formatting of the data values for the flux, fluence and dose results in the generated output files has been improved. These are now all shown in scientific notation, instead of the fixed precision form.

The dose calculations, using the underlying ShieldDose2 model, were revised in order to produce consistent results. Internally, the dose depth values specified by the user are ‘padded’ with extra depth values to improve the stability of the results; these previously showed variability depending on the number of depths specified, and their relative spacing.

The calculation of the magnetic local time (MLT) and L* values in the Adiabatic output file were revised. The calculation of MLT was revised to use the time at the magnetic equator. The conversion of the Phi value to L* was changed to use a fixed value of the magnetic moment, associated with the J2000 epoch date. The default upper limit of the calculated HMin values was increased to 50,000 km.

The Ae9Ap9Gui application was only slightly changed, adding user controls for the specification of the accumulation time interval that is applied to both the fluence and dose value calculations, and for the number of processors to be used during the Ae9, Ap9 and/or SPM model run calculations. Its configuration file now also supports the use of environment variables for the specifications for filenames and paths.

A new TotalDose post-processing utility application was added to the distribution. This utility produces a new set of output files containing the sum of the electron and proton dose results (when available), and calculates their associated aggregation confidence level results when needed. This corrects an issue with the on-the-fly summation previously performed for the GUI plotting of dose results. The GUI was modified to automatically invoke this utility when dose calculations are requested, and the plotting functions use the new files produced by this utility.

The existing IntegralPlasma post-processing utility application was revised, correcting issues with the Plasma model integral flux aggregation confidence level results, and the handling of uni-directional flux values. This utility is also automatically invoked by the GUI when needed.

Additional Third-Party Library Usage

The use of a Message Passing Interface (MPI) library was added to support the parallelized operation of the Ae9Ap9 model calculations. The open-source *OpenMPI* implementation of this library is used on Linux platforms. The *Intel MPI Library*, a commercial product, is used on *64-bit* Windows platforms; the freely-redistributable run-time files for this library are included in the Ae9Ap9 model distribution, and may be used without cost. See the Ae9Ap9/documents/Licenses/Intel_MPI_license_info directory for more information. Please note that parallelized processing on *32-bit* Windows platforms is not supported.

Changes in Supporting Files

The previous set of ‘console test’ model run input files have been relabeled as ‘samples’; the annotations within these input files have been revised to improve their clarity. The previous set of ‘unit test’ model run input files have been relabeled as ‘old unit tests’; these are present for continuity from the previous releases, but will not be present in any future releases. An entirely new set of ‘unit test’ model run input files have been added; these files exercise the numerous available model parameter permutations more thoroughly. These unit tests will be further expanded in future releases.

Documentation Changes

- The *User’s Guide* document was significantly updated to describe the many new and revised keyword/value parameters, and processing features and capabilities.
- The *Build Instructions* document was updated to include the installation of the OpenMPI third-party software package. Information for building on other Linux distributions was added.
- The *Application Program Interface* document was significantly revised to add the new model-level API method access, and update the application-level API method access. The obsolete API for the C and FORTRAN languages were removed.

Summary of CmdLineAe9Ap9 Input File Changes for v1.35.x

The Old Parameters and the New

Starting with the Ae9Ap9 model v1.35.001 release, the specification for many parameters have been revised or added. These tables below only show those parameter keywords and/or values that have been changed, altered behavior, added or removed. Refer to the User's Guide for their full descriptions. All other parameters are unchanged from previous releases. Many keyword and/or value changes were made to clarify the usage of the parameter setting.

Basic Model Inputs

Disposition	Keyword	Value(s)	Description of Change
Altered	FluxOut	Mean Percentile,## Perturbed,### MonteCarlo,###	Scenario number values '##' may now be specified as a comma-separated list, and/or include ranges. Ie: Perturbed,1-20,25,26,30-40
Changed Altered	FluenceOut FlueOut	True False	'FluenceOut' now also accepted as keyword for improved clarity of meaning. Previous specifications of mean/percentile/etc are replaced with True False; fluence calculations are wholly dependent on FluxOut settings.
Changed Altered	DoseRateOut DoseOut	True False	'DoseRateOut' now also accepted as keyword for improved clarity of meaning. Previous specifications of mean/percentile/etc are replaced with True False; dose rate calculations are wholly dependent on FluxOut settings.
Changed Altered	DoseAccumOut CDoseOut	True False	'DoseAccumOut' now accepted as keyword for improved clarity of meaning. Previous specifications of mean/percentile/etc are replaced with True False; dose accum calculations are wholly dependent on FluxOut settings.

Advanced Model Inputs

Disposition	Keyword	Value(s)	Description of Change
Changed	AdiabatOut OutputAdiabat	True False	Keyword changed to be consistent form with other parameter keywords. OutputAdiabat is deprecated.
Changed	MCEPOCH Epoch	<time value>	Keyword changed to specifically identify it being connected to MonteCarlo (and not ephemeris). Epoch is deprecated.
Changed	ChunkSize PtsPerCall	<value>	Keyword changed to a more generally understood data throughput sizing term. PtsPerCall is deprecated.
Added	NumProc	<value>	New parameter that specifies number of processors to be used during parallelized model calculations
Added	AsciiOutput	All None etc	New parameter that allows user control for the selective generation of ASCII output files from the temporary binary files.
Added	WorkDir	<path>	New parameter that allows user control for location of the directory of temporary binary files
Added	BinDirName	<path>	New parameter that allows user even finer control for directory of the temporary binary files
Added	DelBinDir	True False	New parameter that allows user control for deposition of the directory of the temporary binary files after completion of the model calculations.

(Accumulation and) Aggregation Inputs

The terms used have been revised to more accurately describe the calculation results.

Accumulations describe the sum and/or average of values over time; *aggregations* describe statistical results from an aggregate of like data (scenarios). The results from the aggregation statistical calculations are now termed ‘*confidence levels*’ (instead of ‘*percentiles*’).

Disposition	Keyword	Value(s)	Description of Change
Added	AccumMode	Cumul Interval etc	New parameter to specify the accumulation mode(s). (previously, the ‘interval’ accumulation mode was implied when ‘DoseIntrvl’ was greater than zero)
Changed Altered	AccumInterval DoseIntrvl	<days>	Revised keyword for accumulation interval for Dose calculations, and now also affects Fluence calculations. DoseIntrvl is deprecated.
Added	AccumIntervalSec	<seconds>	Same as AccumInterval, but with units of seconds.
Added	AccumIncremSec	<seconds>	Time increment used with <i>experimental</i> ‘Boxcar’ accumulation mode
Added	AccumIncremFrac	<fraction>	Same as AccumIncremSec, except expressed as a fraction of the AccumInterval[Sec] duration.
Added	Aggregate	Mean Median Percent, ###	New keyword replaces all previous data-specific keywords (ie PMAggX, MonteAggCD, etc). Performs aggregation calculations on <i>all</i> Perturbed and/or MonteCarlo scenario results generated. The ‘mean’ of an aggregation has indeterminate meaning. The ‘percent’ (instead of ‘percentile’) confidence level range is now 0-100, with some caveats.
Removed	<i>PMAggX, PMAggF, PMAggD, PMAggCD, MonteAggX, MonteAggF, MonteAggD, MonteAggCD</i>	<n/a>	All data-specific aggregation keywords have been superseded by new ‘Aggregate’ keyword

Dose Calculation Inputs

Disposition	Keyword	Value(s)	Description of Change
Changed	DoseDepthUnits <i>DoseDepthU</i>	millimeters mm mils gpercm2	Revised keyword to spell out ‘units’; added ‘mm’ to acceptable value list. DoseDepthU is deprecated.
Changed	DoseGeometry <i>DoseDetGeom</i>	<geom>	Revised keyword to eliminate misnomer – there is no geometry of the <i>detector</i> . DoseDetGeom is deprecated.
Changed	DoseDetector <i>DoseDetType</i>	<type>	Revised keyword to spell out ‘detector’ for clarity. DoseDetType is deprecated.
Changed	DoseAttnMode <i>DoseAttnMd</i>	<mode>	Revised keyword to spell out ‘Mode’ for clarity. DoseAttnMd is deprecated.
Added	ShieldDose2DB DoseModelDB	<path/file>	New keyword to allow explicit specification of dose model database, if needed.
Removed	<i>DoseIntrvl</i>	<days>	Superseded by AccumInterval.

Orbit Propagation Inputs

Disposition	Keyword	Value(s)	Description of Change
Changed	OrbElemTime <i>OrbElmTim</i>	<time>	Revised keyword for clarity. OrbElmTim is deprecated.
Changed	OrbIncl <i>OrbInclin</i>	<value>	‘OrbIncl’ keyword is now also accepted.

Changed	OrbRAAN OrbRtAsc OrbRAAsNd	<value>	Revised keyword for clarity, with two acceptable forms. The cryptic OrbRAAsNd is deprecated.
Changed	OrbEcc OrbEccen	<value>	'OrbEcc' keyword is now also accepted.
Changed	OrbMeanAnom OrbMeanAn	<value>	'OrbMeanAnom' keyword is now also accepted.
Changed	OrbMeanMot OrbMeanMo	<value>	'OrbMeanMot' keyword is now also accepted.
Changed	Orb1stDerMM Orb1stDer	<value>	Revised keyword for clarity. Orb1stDer is deprecated.
Changed	Orb2ndDerMM Orb2ndDer	<value>	Revised keyword for clarity. Orb2ndDer is deprecated.
Changed	OrbAltApog OrbAltApo	<value>	'OrbAltApog' keyword is now also accepted.
Changed	OrbSemiMaj OrbSmjAxis	<value>	Revised keyword for clarity. OrbSmjAxis is deprecated.
Changed	OrbLTApog OrbLocTimeApo	<value>	Revised keyword for clarity. OrbLocTimeApo is deprecated.
Changed	OrbLTMaxIncl OrbLocTimeMaxIncl	<value>	Revised keyword for clarity. OrbLocTimeMaxIncl is deprecated.
Changed	OrbTimePer OrbTimPerig	<time>	Revised keyword for clarity. OrbTimPerig is deprecated.

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

Disposition	Keyword	Value(s)	Description of Change
Added	FluxOut FluenceOut DataRateOut DataAccumOut	True False	Revised legacy model parameters to use same keywords as Ax9 for data specification. OutData is deprecated.
Removed	OutData	Flux, Fluence, etc	Superseded to use same keywords as Ax9 for data specification (ie FluxOut, FluenceOut, etc). OutData is deprecated.
Changed	LegActLevel REActLvl	<value>	Revised keyword for clarity. REActLvl is deprecated.
Changed	LegActRange REActRange	<value>	Revised keyword for clarity. REActRange is deprecated.
Changed	Leg15DayAp RE15DayAp	<value>	Revised keyword for clarity. RE15DayAp is deprecated.
Changed	LegFixEpoch REFixEpoch	True False	Revised keyword for clarity. REFixEpoch is deprecated.
Changed	LegShiftSAA REShiftSAA	True False	Revised keyword for clarity. REShiftSAA is deprecated.

Appendix B: Legacy CAMMICE/MICS Model Inputs

Disposition	Keyword	Value(s)	Description of Change
Changed	CamMFModel CIModel	<value>	Revised keyword for clarity. CIModel is deprecated.
Changed	CamDstData CIDDstData	<value>	Revised keyword for clarity. CIDDstData is deprecated.
Changed	CamSpecies CISpecies	<species>	Revised keyword for clarity. CISpecies is deprecated.
Changed	CamPAngle CIPAngle	<PA bin>	Revised keyword for clarity. CIAngle is deprecated.

Changes in the output file-naming scheme used in CmdLineAe9Ap9

The names of the output files generated by the CmdLineAe9Ap9 application are constructed based on the input file parameter settings, as described in the ‘Model Output Files’ section of the User’s Guide. The names are built as a sequence of specific words that are determined based on the output file ‘prefix’ name, the respective data mode of the flux model calculation, the type of data result, as well as additional processing and/or further specifications.

V1.30 and before:

Prefix	Data Mode based on <*Out> value	Data Type based on <*Out> keyword	Percentile, Scenario and/or Aggregation Id, based on <FluxOut> and <*Agg*> keywords	Suffix
<OutFile> (without the filename extension, ie ‘.txt’)	mean	<i>flux</i> <i>fluence</i> <i>doserate</i> <i>totaldose</i>	(-n/a- for mean)	filename extension of <OutFile> or ‘.txt’ if not included
	_pctile		_## (percentile, in <FluxOut> value)	
	_pert		_### (scenario identification #)	
	_mc		_agg_mean <i>_agg_median</i> <i>agg_pctile</i> ##	

In v1.30 and before, the use (or absence) of the ‘DoseIntrvl’ accumulation specification parameter had no effect on the dose output filenames that were generated.

In v1.35, the ‘DoseIntrvl’ parameter has been superseded by the ‘AccumInterval’ parameter, where this new parameter now applies to the Fluence results, as well as the Dose results. This accumulation interval value may be augmented with the ‘AccumMode’ parameter for specifying how the accumulation calculations are performed, and more than one mode can be specified.

Those ‘Data Type’ filename segments shown in the blue text, have been revised in v1.35 to now also reflect the Accumulation mode of the data contained within:

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	-----	-----

Also note that the ‘totaldose’ misnomer has been replaced with ‘doseaccum’. The filenames now reflect the accumulation mode (but do *not* specify the actual accumulation interval durations as part of the filename).

The ‘TotalDose’ post-processing application will read the existing electron and proton model ‘doserate’ and/or ‘doseaccum’ output files, and produce new ‘Totaldoserate’ and/or

‘Totaldoseaccum’ output files that contain the combined the electron and proton dose information (these files will use the OutFile prefix specified in the electron model input file).

The ‘_agg_pctile_’ and ‘_agg_median’ filename segments, shown in orange, have been replaced with ‘_conf_level_’, reflecting the terminology change for the aggregation statistical calculations. The output files for the ‘mean’ of scenarios may be produced, using the ‘_agg_mean’ filename segment, but these calculation results are of indeterminate meaning, and their use is discouraged. The ‘mean’ is *not* confidence level.

Overview of the Parallelized Model Calculation Operations

The previous releases used a CmdLineAe9Ap9 application that was rather monolithic, in that all calculations were performed through this program only. Although an API was available, the types of operations offered were severely limited, mainly the ephemeris generation and raw flux calculation; operations for dose calculations were not independently available.

Parallelization of these model calculations required the dissolution of this monolithic approach, separating the calculations into wholly independent components: Ephemeris, Flux, Fluence, Dose and Aggregation. The existing code files were examined and reorganized accordingly. Mirroring the current 'Ae9Ap9Model' class, other new classes were created for the other components, ie 'EphemModel', 'DoseModel', etc. Some of these became simply a wrapper upon their supporting SpWx library classes.

In order to properly generate and maintain control of the potentially large amount of calculations, the writing and reading of files containing these data is required. To maximize efficiency, during the various model processing steps, all data are maintained in straight native binary-format files; these files are converted to their equivalent ASCII-format files, when required, only at the conclusion of the processing. To avoid directory file pollution, these binary files are written to/read from a uniquely-named temporary directory. An optional parameter allows the location of this temporary directory location to be specified by the user, for possibly improving file I/O performance. To handle this file management, several classes were developed for the reading and writing of both binary and ASCII forms of datafiles, the generation of a standardized file header format and a dynamic file-naming scheme.

A new level of program classes was created for the generation of the files containing the calculated data, interfacing with both the model-specific classes and the file I/O classes. The specific task executors then interface with this new level of classes. The top-level CmdLineAe9Ap9 program was redeveloped as the task ‘scheduler’, determining which task executors were required based upon the model run input file parameters. This queue of tasks is used by the ‘scheduler’ to spawn the necessary ‘executor’ task program once the previous task was complete. In single-threaded mode, the executor program is provided with the information needed to complete its task. In multi-threaded mode, the scheduler provides the needed information via MPI-based communication. The scheduler also receives occasional updates from each of the executors, so to be able to calculate the overall progress of the entire model run, written to a progress file. Each of the task executors also read the model run input file, extracting only the parameter settings relevant to their specific task functions.

The calculation of the flux values is likely the most computationally-intensive portion of the typical Ae9/Ap9/SPM model run, especially when dealing with large numbers of ‘scenarios’. It was determined that the most effective parallelization of such calculations would be to break up a model run by time (or position entries), into a number segments that equaled the number of

processing nodes available (one less than the total number of processors, as the ‘scheduler’ application requires a dedicated processor). To accommodate this segmentation, the first executor task processes the ephemeris information, whether input or generated, dividing this into separate segments of equal number of entries and written to their respective binary files. Any specified direction vector and/or pitch angle information is merged into these files at this time.

After the completion of the ephemeris processing step, their segmented binary files are used as input to the next executor task, for the flux calculation. Within here, the adiabatic invariant values are calculated and then used for all requested ‘use case’ flux values (excluding aggregations), writing to their respective binary output files.

To make the subsequent processing of the generated flux values more straight-forward, the binary file segments of each ‘use case’ flux values are concatenated together using the ‘concat’ task executor. The use of straight binary files (as opposed to organized ones, such as HDF5, NetCDF, etc) enables such files to be simply appended. This task executor is always performed by a single thread, as it is primarily file I/O operations being performed. Research has shown that multiple programs simultaneously performing file I/O on the same disk drive will *significantly* degrade the disk performance; RAID-5 disk units may be the only exception to this condition.

If the fluence and/or dose results are requested, those files of full time period flux values are processed by the ‘fluedose’ task executor. Because the accumulation time (if specified), fluence and dose calculations are very much intertwined, it is more effective to perform these functions in tandem. In addition to identifying the type of data, the generated output filenames also relate the type of accumulation applied. When performing this task in parallelized fashion, each available node is used to process one of the input flux files; the ‘scheduler’ communicates the name of the next input flux file whenever a node completes its processing, until none remain.

The ‘aggreg’ task executor is utilized when the aggregation of scenarios is requested, processing the sets of files in a similar fashion to the ‘fluedose’ executor.

The final ‘convert’ task executor is invoked for the conversion of all completed binary files into their corresponding ASCII form. This executor is also performed by a single thread, for the same reason as explained for the ‘concat’ executor.

At the conclusion of all processing, the ‘scheduler’ process automatically removes all intermediate binary files generated during the model run execution, unless an optional parameter setting is specified to retain them. The format of these binary files mirrors the contents of their ASCII equivalents, with the exception that the time is always stored as a Modified Julian Date value.

Contact Information

Please send any questions, comments and/or bug reports to: ae9ap9@vdl.afrl.af.mil

The 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>