

SIMMOD Source Code Change Descriptions

Notes on version numbering:

After the release of version 2.2.23.11, ATAC changed the version numbering philosophy such that released executables were always numbered 2.2.x. Consequently, version 2.2.23.11 was considered to be version 2.2.24 within ATAC. Beginning with version 2.3, all major releases were numbered 2.x and minor releases are 2.x.y. Both major and minor releases were placed within the configuration management and version control system. The “z” identifier in 2.x.y.z was used for test versions only. Test versions were not always placed under version control.

In May 2004, the numbering of the ATAC engine was synchronized with the version numbering of the Simmod *PLUS!* and *PRO!* systems. Consequently, version 2.6.1.4 was followed by version 6.2.0.

Later revisions contain all the changes from the previous revisions.

SIMMOD Version	Change Item	Change Description
2.2.24.1	1	Changes the serial number of the departure segment of an “arrival→departure” turnaround flight from that of its arrival segment in the SIMU26 file. These two serial numbers were previously the same. All flight segments now have a unique SIMU26 serial number. The two segments of continuation flights (departure→arrival) currently have different serial numbers.
2.2.24.2	1	Corrects the hold duration for the FH code in the SIMU26 file. This was previously incorrect in some cases. Also eliminates some HA/FH entry pairs when the HA/FH pair results in a zero hold duration.
	2	Allows wild cards at the beginning and end of the names in the route list of the SETCLONE input. *RTE* matches any route name that contains RTE (new option). RTE* matches any route name that begins with RTE (this option previously existed). *RTE matches any route name that ends with RTE (this option previously existed but was undocumented).

- 1 Enhances the procedure logic. A procedure end node and a touch-down distance can be specified in addition to the existing procedure start node. These two new inputs are applied somewhat differently for arrival and departure procedures.

Arrivals

The logic assumes that the arrival interface node is located at the arrival end of the runway (i.e., co-located with the first/last node of the runway). The actual point at which an aircraft touches down is determined by taking the greater of: (a) the runway displacement threshold, as defined for the given runway in the RUNWAYS input, (b) the touch-down distance, as defined for the given arrival procedure in the PROCEDURES input, or (c) the sum of the runway link lengths from the arrival end of the runway to the procedure start node, as defined for the given arrival procedure in the PROCEDURES input.

Given these enhancements, the best method of creating an arrival procedure is to specify the arrival end of the runway as the procedure start node with an appropriate touch-down distance.

Runway nodes not bounded by the procedure start and end nodes are *not* preempted by the landing aircraft unless that aircraft *must* use those nodes for its rollout as determined by its touch-down and roll distances. For example, an analyst can specify that an arrival procedure start and end at the same node. This arrival will then preempt the nodes from the arrival end of the runway to its runway exit node since it will either fly over or roll on these nodes. Under these conditions, taxiing aircraft will *not* be prohibited from crossing the runway at any node past the landing aircraft's exit node since such a node is not included in the arrival procedure and is unnecessary for the landing/rollout.

Departures

For most departures, the first node of the procedure should be the point at which the aircraft enters the runway to begin its takeoff roll. The last node of the procedure should be the node of the runway beyond which runway nodes are *not* blocked for taxiing. Note that the departure interface node should be co-located at the last procedure node since this is the location at which the aircraft is entering the airspace system. For example, if it is desired that the departure block the rest of the runway while overflying it, the last node of the runway should be the last node of the procedure as well.

	2	Adds a CN code to the SIMU26 file to indicate the change of serial numbers of turnaround aircraft. The format is as follows: CN <time of turnaround> <old serial number> <new serial number>										
	3	Corrects landing roll, in which first reservation of a landing roll was not being removed properly.										
2.2.24.4 1997-04-22	1	Corrects the MULTIFIT control logic. An aircraft could previously be given an additional vector delay (equal to the maximum for the link) every time another aircraft was placed before it in an airspace node arrival queue. These additions would be made regardless of the amount of vector delay the aircraft was already incurring at that time.										
2.2.24.5 1997-05-24	1	Corrects high-speed exit logic. Previous logic allowed high-speed exits, even if the landing roll distance was not satisfied. Also, aircraft were able to exit on links that required turns of greater than 90 degrees off the runway. High speed exits are allowed under the following turning limits: <table border="1" data-bbox="527 877 1339 1165"> <thead> <tr> <th>If the difference between the runway and taxiway headings is $\Delta\theta$</th> <th>The aircraft must have traveled this percentage of its total landing roll</th> </tr> </thead> <tbody> <tr> <td>$0^\circ \geq \Delta\theta > 10^\circ$</td> <td>80%</td> </tr> <tr> <td>$10^\circ \geq \Delta\theta > 20^\circ$</td> <td>85%</td> </tr> <tr> <td>$20^\circ \geq \Delta\theta > 30^\circ$</td> <td>90%</td> </tr> <tr> <td>$30^\circ \geq \Delta\theta > 90^\circ$</td> <td>95%</td> </tr> </tbody> </table>	If the difference between the runway and taxiway headings is $\Delta\theta$	The aircraft must have traveled this percentage of its total landing roll	$0^\circ \geq \Delta\theta > 10^\circ$	80%	$10^\circ \geq \Delta\theta > 20^\circ$	85%	$20^\circ \geq \Delta\theta > 30^\circ$	90%	$30^\circ \geq \Delta\theta > 90^\circ$	95%
If the difference between the runway and taxiway headings is $\Delta\theta$	The aircraft must have traveled this percentage of its total landing roll											
$0^\circ \geq \Delta\theta > 10^\circ$	80%											
$10^\circ \geq \Delta\theta > 20^\circ$	85%											
$20^\circ \geq \Delta\theta > 30^\circ$	90%											
$30^\circ \geq \Delta\theta > 90^\circ$	95%											
	2	Added two new optional fields to the end of the RUNWAYS and SETRWY data records. ADistNoO specifies the landing roll distribution and DDistNoO specifies the takeoff roll distribution, respectively, for the runway in the opposite direction. If these fields are omitted, the distributions for the rolls in the primary direction are used for the opposite direction to ensure compatibility with older input files. The fields for the distributions in the primary direction have been renamed to ADistNoP and DDistNoP. If values for ADistNoO and DDistNoO are entered, values for ADistNoP and DDistNoP must also be entered.										
	3	Restored touch-and-go logic from the version of SIMMOD 2.2 that was received from the FAA (transfer from CACI).										
2.2.24.6 1997-07-22	1	Corrected the procedure input routine such that the TDDist field is optional.										

- 2 Corrected GATERWY logic such that a GATERWY entry that designated a taxipath for both arrivals and departures is now selected. Previously, such taxipaths were ignored.
- 3 Removed “Demo” and “Beta” version checking.
- 4 Disabled logic that exempted aircraft from air holding if the hold is expected to be less than or equal to 0.75 seconds.
- 5 Corrected the message in entity/set pointer range check.
- 6 Assigned a trace message to trace number 340 from 000.
- 7 Modified slightly the last node departure procedure logic described as item 1, version 2.2.24.3. Still works as described.

2.2.25
1997-09-19

- 1 Added a field to the end of the RT and RL codes in the SIMU26 file. This field provides the aircraft speed (in nautical miles per hour) during these movements.
- 2 Added logic and routine to print the SIMU40 file that provides additional input echoes.

- 3 Renamed variables to improve the naming convention and compile-time checking.

All variables are now explicitly declared as opposed to the implicit declaration used previously. This uncovered a few bugs in which the SimScript compiler was not detecting longstanding typographical errors but instead assumed they were integer variables.

The scope of a variable can now be determined by simply observing at the variable name. Periods (“.”) are used in only the following ways:

Global variables and arrays begin with two periods.

example: `..ALL_SPDS_ARE_TRUE`

Local variables and arrays begin with period.

example: `.RTE`

Entities begin with a prefix followed by a period then the attribute. This is consistent with the way class member variables are referenced in C++ and Java.

example: `DQ.APT_NO`

Compound entities have prefixes delimited by periods.

example: `DQ.IFN.DEP_PRI_FLG`

Arguments passed into a routine have no periods in the called routine.

All other periods have been converted to underscores. Various names have been changed for improved readability.

- 4 Corrected the arrival spread logic. If the spread values were denoted as a percentage of the node separation, the affected departure queues were not deactivating the spread imposition once the gap time elapsed or the threshold was no longer exceeded. The proper termination of spread imposition has been enabled. In addition, the spread percentage was being applied to the wake turbulence separation as well as the node separation. This has been corrected such that the spread percentage applies to the node separation only.
- 5 Changed logic to read airlink lengths as real values. These values are still in tenths of miles to maintain backward compatibility.
- 6 Changed all single precision real values to double precision. Previously, there was a mix of single precision and double precision real variables throughout the code. Comparisons between single and double precision variables sometimes provide unexpected results.

- 7 Added departure logic to optionally allow an aircraft to be released from a departure queue only if a previous departure on the same runway is airborne, regardless of the procedure time and distance separations between the two departures.
- 8 Added logic to allow the user to specify whether or not a taxiing aircraft can cross a runway in front of a departure when the departure is airborne.
- 9 Modified the runway crossing logic. Corrected bugs in which runway crossing aircraft were not accumulating hold delay properly nor switching the priority at the correct time. Implemented logic that enable at least one aircraft at each of the defined RWYCROSS points time to cross once one of the waiting aircraft has switched the priority for runway crossing.
- 10 Enabled lateness and de-icing inputs for turnaround flights.
- 11 Corrected a bug that prevented CONCOURSE inputs from exceeding a single line in the input file. Each CONCOURSE input can now extend to additional lines as long as they do not start in the first column.
- 12 Corrected a bug that prevented the holdcycle field from being read properly in the CHECKPOINT input.

2.2.26
1997-11-26

- 1 Minor corrections in TRACE messages.
- 2 Added logic and routines to write out SIMU42 file that contains sector entry and exit information.
- 3 Added SF history code to indicate the completion of staging with the duration.
- 4 Corrected a bug in the takeoff logic, in which aircraft taking off did not properly check that the runway was being blocked by arrival aircraft. (This bug was not addressed during the implementation of Item 7 of version 2.2.25.)

2.2.26.1
1998-02-16

- 1 Corrected a bug in which a landing aircraft will not immediately enter a ground link but, instead, wait in the link's queue behind other aircraft, even if the landing aircraft is getting off a runway. That is, landing aircraft should have priority on entering ground links when it results in their exiting the runway. This was not the case when other arrival aircraft (that had previously landed) are waiting to enter the link.

- 2 Corrected logic in which the taxi delay time is computed. Incorrect calculations were occurring when an aircraft had to wait to cross an active runway.
- 3 Corrected logic in which departures wait indefinitely in the departure queue in cases where there are multiple runway crossing points and more than one runway crossing aircraft imposes crossing priority simultaneously.
- 4 Corrected the duration, link length, and speed delay fields of the air movement codes (NM, SU, SL, or PS) of SIMU26.
- 5 Added RE history code to indicate that an aircraft changes its arrival time to the next airspace node. A new air movement code (NM, SU, SL, or PS) immediately follows the RE code to replace the aircraft's last movement code.
- 6 Added optional global data input DELAY_CALCULATION to allow user control over how an aircraft speed up (positive deviation from nominal speed) is tabulated in the air travel delay field of SIMU14.
- 7 Improved ground node/runway assignment. Previous logic permitted a ground node to be aware that it was on a runway but would only record the last runway to which it was assigned (via the RUNWAYS definitions in SIMU07). Each ground node now maintains a list of all the runways to which it belongs.
- 8 Removed default of 3.0 nm setting for wake turbulence separations. Previously, any value of 0.0 in the wake turbulence separations list of the AIRCRAFT definitions in SIMU03 was automatically set to 3.0.
- 9 Added an optional global data input keep_window_active to allow user control to keep the simulation window open when the simulation ends. When the option is activated, the user must press the <ENTER> key close the window and terminate the application.
- 10 Corrected taxi planning logic to allow U-turns on an active runway by departing aircraft.

2.2.27
1998-06-29

- 1 Corrected an error in the gate selection logic. In the event of a missed approach, a destination gate is rechecked for availability. Previously, this recheck could crash the simulation.

-
- 2.2.27.1
1998-07-01
- 1 Corrected logic that determines the runway landing roll exit. Previously, the logic did not check the directionality of candidate exit links. Consequently, a link with a directionality opposite to that of the landing aircraft could be chosen as the anticipated initial link of the taxipath. This could lead the taxi planning logic to fail if no other exit links are available and taxiing on the runway is disallowed. Also, proper checks have been added to determine if non-duty runways are available for taxiing.
 - 2 Added logic to try to launch a departure after a landing aircraft crosses a runway node. This should help activate aircraft waiting to take off from a crossing runway.
 - 3 Corrected logic described in item 7 of version 2.2.25. Previously, the logic would allow a departure to takeoff once another departure on an intersecting runway becomes airborne. The current logic now requires that the prior departure clear the runway intersection, regardless of whether or not it is airborne, before the subsequent departure is cleared to takeoff.
 - 4 Corrected plan change logic. Previously, if a plan change occurred after an arrival event was processed but before the aircraft was injected (possible if there is a non-zero lateness), the arrival would attempt to fly the incorrect route. Also, the logic performed improper checks that would result in simulation aborts.
 - 5 Added a global input `airport_dq_pass_depth` for airport-wide departure queue passing. An optional global parameter has been added that limits the depth to which the “next departure” logic searches the airport-wide departure queue for prospective departures. Until this change, all the aircraft in the airport-wide departure queue were examined often resulting in one departure queue dominating the others at airports with multiple departure queues. If this parameter is set to zero, only that departing aircraft that has been waiting the longest in a departure queue at the airport is permitted to depart. If missing, all departures are examined. The passing depth parameter specified for each departure queue is unaffected and may still limit the depth in the queue that departures are examined.
 - 6 Corrected a Year 2000 bug. Replaced calls to the faulty SimScript `DATE.R` routine with calls to a new routine that uses an ANSI C date/time function.

2.2.27.2 1998-07-17	1	Corrected runway crossing logic. A primary crossing aircraft can set the priority for all the RWYCROSS points on a runway once that primary aircraft's thresholds have been met. However, the previous logic allowed this primary aircraft to set the priority on a "secondary" runway crossing if an aircraft was actively crossing but not waiting at that crossing point. This could result in the secondary RWYCROSS having priority with no aircraft waiting, yet departures would be held up.
2.2.27.3 1998-08-25	1	Added new global setting max_num_final_arrival_links. This variable contains the maximum number of links prior to the interface node that an arrival aircraft will begin procedure blocking. Defaults to 2. (e.g., an aircraft will not begin blocking related procedures until it is no more than two links away from the arrival interface node.) EEC
	2	Changed logic such that if an aircraft cannot proceed to its next node because it cannot enter a full sector it no longer prevents other aircraft that are not subject to the "full sector" constraint from proceeding to that node. EEC
	3	Corrected stagger logic. Previously, aircraft could sometimes be released prematurely from stagger separation holding. EEC
2.2.27.4 1998-11-05	1	Corrected route separation calculation. Previous logic did not include the takeoff roll distance of a leading departure when determining the distance that the leading departure is ahead of a following departure. Consequently, the leading departure traveled farther than specified by the route separation input before permitting the following departure to takeoff.
	2	Enhanced the usage of the QI and QO codes in the SIMU26 history file. These codes are only used when the use_depq_hold_codes global input is set to 1. The QI code is substituted for the H3, H4, and H7 codes when the reason a taxiing aircraft must hold is entirely due to downstream traffic in which all the intervening aircraft are also destined for departure queues. Likewise, the QO code is substituted for the F3, F4, and F7 codes.
2.2.28 1999-03-26	1	Changed towing logic. The towing speed factor, tow_spd_fc, global input was not being used when calculating ground link transit times for towed aircraft. This has been corrected. Logic has also been added to allow a random selection of towing areas. EEC

- 2 Fixed staging logic. Aircraft leaving the staging area were not removed from the gate queue until arriving at the gate. This led to problems in the routine that removes an aircraft from the staging pad that has already left for the gate. **EEC**
- 3 Added the SETSEP event. This event allows wake turbulence separations to be changed during the simulation. **EEC**
- 4 Changed wake turbulence logic. Previously, the logic would use a trailing aircraft's speed in the calculations in the earliest time-of-departure and time-of-arrival at an airspace node. A new global data input node_sep_ac_choice has been added to provide a choice as to whether the leading or the trailing aircraft's speed should be used. **EEC**

2.2.29
1999-06-11

- 1 Fixed procedure logic. A departure procedure would not properly block another departure procedure on a different runway when the second aircraft was departing simultaneously or immediately after the first.
- 2 Fixed an error code for printing trace message 299 that resulted in engine crashes under certain conditions.
- 3 Improved the logic so that the input and output files are closed properly when a simulation aborts.
- 4 Improved takeoff and landing logic. Previously, this logic not did consider the effects of wind. Now, the groundspeed of an aircraft is properly calculated during takeoff and landing rolls.

2.3
1999-10-15

- 1 Fixed an error in the taxi logic. The engine would crash in specific cases where an aircraft arrived at an occupied gate. This has been fixed.
- 2 Fixed an error in the runway occupancy logic. In cases where runway reservation blocking by a trailing arrival would begin before a preceding arrival reaches the interface node, the preceding arrival would interpret the trailing arrival's runway reservations as a fouled runway.
- 3 Changed taxi logic. Previously, an aircraft was allowed to enter a ground link that was filled to capacity if all the aircraft on that ground link were traveling in the opposite direction and the ground link was not in a DSDPATH. Now, this logic is active only if the new global data input gnd_link_capacity_override has been set to 1.

- 4 Modified DSDPATH logic. A few small modifications were made to this logic to better determine when an aircraft can and cannot enter a DSDPATH. This should reduce some excessive ground delay in certain situations.
 - 5 Fixed a bug in the simulation termination logic. In order to successfully terminate a simulation, all of the link list data structures must be emptied before the objects in them are destroyed. In cases where an aircraft was in a holdcycle waiting for an available gate, the aircraft object was not being removed from a "holding list" properly.
 - 6 Modified the taxi logic. Under certain conditions when entering or taxiing on a DSDPATH, aircraft would not be separated properly. Checking logic modified to better detect such occurrences.
 - 7 Enhanced checkpoint logic. A new, optional integer field was added to the CHECKPOINTS inputs. If the ReEval field is set to 1, the taxiing arrival will re-evaluate its gate selection. If its assigned gate is blocked, the aircraft will re-examine all the gates for a possible reassignment and then go to staging (or a holdcycle) if staging pads (or holdcycles) are defined for its current gate assignment and no other gates are available. If the ReEval is missing or set to 0, no re-evaluation will be made upon reaching a checkpoint.
 - 8 Modified bank logic. Under certain conditions (esp. turnarounds?), aircraft were not filed into a bank properly. This has been fixed.
- TransSolutions**
- 9 Modified the reporting of de-icing events. Under certain conditions, the DI code was written incorrectly to the SIMU26 file. The taxi out time was also not accumulated properly. This has been fixed. **EEC**

2.3.1
2000-06-21

- 1 Modified taxi logic. Previously, aircraft rolling on runway (and high speed exit) links during takeoff and landing rolls would not be included in ground link occupancy counts. Such aircraft are now counted.
- 2 Enhanced checkpoint logic. Previously, it was necessary to define staging areas if a default checkpoint was defined for a gate. Staging areas no longer need to be defined for the logic described in change 2.3:1 to function properly.

- 3 Enhanced landing logic. Previously, the landing logic assumed that the arrival interface node would be located at the arrival end of the runway. The new logic assumes that the arrival interface node is located at the arrival procedure start node which may be any node on the runway. If the procedure start node is “upstream” of a defined runway threshold, the aircraft will overfly a sufficient distance along the runway links so that it touches down at the displaced threshold. If a procedure touch-down distance is defined, the aircraft will overfly that amount of ground link distance from the procedure start node. Any overflight of the runway due to a displaced threshold will concurrently satisfy that amount of a defined touch-down distance and vice versa.
- 4 Changed the effect of the `use_depq_hold_codes` global input. When `use_depq_hold_codes` is set to 1, a QA (instead of a QI) is substituted for the H3, H4, and H7 codes as previously described in change 2.2.27.4:2. A QB (instead of a QO) code is now substituted for the F3, F4, and F7 codes.

When the `use_depq_hold_codes` global input is set to 2, QI and QO codes are written to the `SIMU26` history file. Each time a departing aircraft must hold while taxiing to the departure queue, all the links in its taxiplan between its current ground node and the departure queue are checked to determine if they are filled to capacity. On the first occurrence when this is true, a QI code is written for that aircraft (in addition to the appropriate hold code). If no QI code is written while the aircraft is taxiing, one is written when it enters the departure queue. A QO code is written when the aircraft exits the departure queue. Consequently, a single QI/QO code pair is written for each departure flight. If the ground link is not filled but is part of a DSDPATH that is filled, the ground link is considered to be filled to capacity.

A value of 3 will activate the use of both the QA/QB and QI/QO codes.

- 5 Enhanced DSDPATH logic. Previously, a ground link could be assigned to only one DSDPATH. Now, a ground link may be assigned to any number of DSDPATHs. Minor changes were made to reduce the likelihood of head-to-head conflicts.
- 6 Added `AFLINK_SPEED_TYPES` input. This set of inputs allows the analyst to group airfield links into airfield link types. For each airfield link type, the analyst can then specify a taxi speed for any combination of aircraft model and arrival/departure type. Any such defined speed supersedes the airfield link taxi speed specified in the `AFLINK` inputs.

- 7 Fixed a bug in the procedure logic. Previously, an arrival would not activate its procedure logic if it was more than two links away from its interface node, regardless of the value of the global variable `max_num_final_arrival_links`. This is now working correctly.
 - 8 Changed SETPLAN logic. Previously, the `PrePlanTime` field must either be missing or greater than the `Time` field. Now, the `PrePlanTime` value is permitted to be the same as the `Time` value, in which case the logic functions as if it were missing.
 - 9 Fixed a bug in the gate selection logic. The engine would crash during the checking imposed by the `GATEUSE` constraints if both the airline and gate needed to be assigned.
 - 10 Changed departure logic. Previously, if an approaching arrival reserves a runway while a departure is taxiing from the departure queue to the runway, the departure will be stopped at its procedure start node on the runway. Now, the departure does not recheck for reservations upon reaching the runway.
 - 11 Fixed tail factor implementation. Previously, the tail factors distribution defined for the aircraft group was not being applied to the node and wake turbulence separations. This has been fixed.
 - 12 Fixed runway crossing logic. In cases where a `RWYCROSS` was defined at an intersection of two or more runways, the logic would consider traffic on only one of the runways. This has been fixed.
 - 13 Fixed stagger bug. The engine would crash in cases where the interval between two staggered aircraft is so great that the leading aircraft is ejected from the simulation before the trailing staggered aircraft reaches the “pre”-stagger node. This has been fixed.
 - 14 Enhanced stagger logic. Previously, arrival interface could not also be defined as stagger node; that is, stagger nodes were required to be upstream of arrival interface nodes. This limitation has been removed. Interface nodes may now also be stagger nodes.
 - 15 Increased the maximum number of allowable plans. The engine architecture requires a hard-coded limit to the number of allowable plans. This was previously set to 5; it is now 50.
 - 16 Added the LA code in the `SIMU26` file. The LA code replaces the RL code for movements in which the landing aircraft overflies the runway.
-

-
- 2.4
2000-10-31
- 1 Added tail factor type. Previously, the tail factor was a percentage increase in aircraft separation. A new field has been added to the AIRCRAFT inputs to allow the definition of a tail factor type. This field can be either PCT or INC. PCT indicates that the tail factors are percentage increases. INC indicates that the tail factors are nautical mile addends. If the new field is missing, PCT is used by default for backward compatibility.
 - 2 Refined tail factor distribution inputs. Previously, only one distribution was used to define the range of tail factors for a given aircraft group. Now, the user has the option of defining a second distribution. If two distributions are defined, the first applies only to arrivals of the given aircraft group and the second applies to departures. If only one distribution is defined, it applies to both arrivals and departures, as before. **TransSolutions**
 - 3 Enhanced banking logic. An optional field has been added to the flight data portion of a BANKS record to designate that the flight is either an arrival or departure. This allows a flight to be entered in the BANKS inputs twice and handled differently depending on whether it is arriving or departing. A second optional field was added to specify the destination of a flight. **TransSolutions**
 - 4 Enhanced arrival procedure selection logic. Previously, if more than one feasible procedure was defined for a plan/interface node/aircraft group combination, the engine would always select the first such feasible procedure. A SIMU07 input, SETCHOOSEPROC, has been added to allow the analyst to specify a distribution of arrival procedures from which an initial selection is made. **TransSolutions**
 - 5 Enhanced runway crossing logic. Previously, a single pair of thresholds for triggering a priority switch to crossing aircraft were specified for an airport. Now each runway possesses its own pair of thresholds which may be modified during a simulation with the SETXNG event. **TransSolutions**
 - 6 Added departure queue holding codes to the SIMU26 file. The codes Q1, Q2, Q3, Q4, Q5, Q6, and Q7 are written for those departure queues whose optional field WaitRsnPrtTF in the DEPARTQ inputs is set to PRINT_HOLD_REASONS.
 - 7 Restored pairing logic. Pairing logic existed in previous versions of the engine. This logic possessed bugs and was disabled in versions 2.2.23.11 through 2.3.1. This logic has been restored and debugged. The optional field Mate in the LINKS inputs has also been restored.

- 8 Improved taxi planning logic. Under certain conditions the taxi planning logic was not properly considering the extra time required for pushback/powerback and would set up a faulty taxi plan. This could lead to gridlock when other taxiing aircraft made decisions based on the faulty plan. This has been fixed for cases where each gate has only one airfield link connected to it; however, given the nature of the logic, there is currently no way to avoid this problem with certainty in cases where a gate has more than one link attached to it.
- 9 Improved DSDPATH logic. Numerous minor improvements made to avoid gridlock conditions.
- 10 Added ground link/ground link blocking. A new SIMU07 input, AFLINKBLOCKING, has been added to allow an analyst to specify that the occupancy of a given ground link by a given ground group (or aircraft model) will block the entry of another (or same) ground link by another (or same) ground group (or aircraft model).
- 11 Fixed airspace logic. Previously, the engine would crash under very specific conditions associated with stagger delays.
- 12 Increased the flexibility of pushback/powerback inputs. A new SIMU07 input, GATE_MDL_PP_TIME, has been added to allow an analyst to specify pushback and powerback time distributions for each combination of gate, gate link, and aircraft model. This input supersedes the PP_TIME inputs.
- 13 Added new output statistics. A new output file SIMU44 has been implemented to provide statistics for each occurrence of the following specific simulation events:
 - De-icing – entry and exit time
 - Departure queue – entry and exit time
 - Staging – entry and exit time
 - Gate – entry and exit
 - Runway crossing – entry, exit, and hold time
 - Multiple runway crossing - entry, exit, and hold time
 - Route – start and end time
 - Runway – entry and exit time
 - Wheels On/Off – time of occurrence

- 14 Enhanced the gate selection logic. A new SIMU07 input, AL_MDL_GATE_CHOICE, has been added to allow an analyst to specify priorities and random selection distributions that are used when a flight chooses a gate. These priorities and distributions are defined for each combination of airline and aircraft model.
- 15 Enhanced the runway exit selection logic. A pair of new SIMU07 inputs, RUNWAY_EXITS_LINKS and RUNWAY_EXITS_PROHIBITED_LINKS, have been added to allow an analyst to specify random selection distributions and lists of prohibited exits for each combination of gate and aircraft model.
- 16 Corrected de-icing/de-staging link capacity violations. Previously, aircraft bound for de-icing/de-staging pads would violate the capacity restrictions for the ground links used to enter the pad nodes.
- 17 Corrected de-icing taxiplan problems. Previously, an aircraft that was required to return to the de-icing pad after reaching the departure queue would either ignore the need to de-ice or formulate an incorrect taxiplan to the de-icing pad. The latter case would result in an engine crash.

2.4.1 2000-11-08	1	Corrected runway exit logic. The runway exit logic new for version 2.4 would not permit a gate to be assigned to more than one concourse, as was previously permitted in other logic. This constraint has been eliminated.
---------------------	---	--

2.4.2 2001-04-18	1	Changes to the AL_MDL_GATE_CHOICE logic. A number of errors in the input logic have been corrected. A semicolon has been added to terminate the first line of a record.
	2	Corrected taxi logic. The implementation of the GATE_MDL_PP_TIME logic introduced a bug, such that the AFLINK_SPEED_TYPES inputs would be disregarded and only the default taxi speed would be used when no GATE_MDL_PP_TIME inputs were defined.
	3	Corrected staging and de-icing input logic. Previously, the records for each combination of plan/gates/pad in the DEPARTQ inputs were implicitly required to be grouped by plan. This is no longer necessary.
	4	Corrected taxi logic. Departing aircraft that were taxiing from a departure queue to a runway procedure start node were prohibited from using ground links that define that runway. This would happen even though the taxi planning logic found that using such links was legal and appropriate.

- 5 Enhanced de-icing pad selection logic. The previous logic would not switch an aircraft to a lower priority pad until the waiting queue of a higher priority pad was filled. An optional field DIStrat has been added to the DEICING inputs. If this field is set to F or left missing, the logic behaves as it did previously. If this field is set to S1 or S2, the logic will switch the aircraft to a lower priority pad if space is available on that pad, even though the waiting queue of the higher priority de-icing pad is not filled. If all the pads and queues are full, S1 will cause the logic to send the aircraft to the highest priority pad and S2 will choose randomly among all the possible pads without regard to priority.
- 6 Corrected takeoff logic. Aircraft performing U-turns on a runway were ignored by aircraft taking off. Now, aircraft performing U-turns are recognized as taxiing.
- 7 Modified tail factor logic. Previously, in cases where there was specifically a distance separation of 0.0 between two related departure procedures, the logic would still apply any tail factor addend. Now, a distance separation of 0.0 implies that no dependency exists between the two departure procedures; consequently, the tail factor addend is not applied in such cases.
- 8 Modified taxi logic. Under certain unusual conditions, the logic would erroneously delay an aircraft from entering a DSDPATH. This has been fixed.
- 9 Improved DSDPATH logic. Minor improvements made to avoid gridlock conditions during certain cases when pushback blocking is being used.
- 10 Improved taxi planning logic. The logic was not properly calculating the time required to taxi on runways. This could result in an aircraft arriving at each ground node of its taxiplan earlier than expected, which, in turn, causes problems later in the DSDPATH logic.
- 11 Fixed the procedure logic. Under certain conditions in cases where arrival procedures begin blocking more than two links from a runway, some of the related departure procedures were not being blocked properly. This is now working correctly.
- 12 Corrected runway exit logic. The runway exit logic would fail to consider U-turns for landing aircraft using runways in the opposite direction. This is now working correctly.

13 Corrected takeoff logic. Aircraft would perform U-turns with no elapsed time when taking off in the opposite direction of a runway. Now, such departing aircraft allow the correct U-turn duration to elapse.

2.5
2001-09-28

- 1 Corrected AFLINKBLOCKING logic. Internal memory arrays and variables were not used properly. Runway links would block other ground links for improperly long times and various blocking combinations would not occur.
- 2 Corrected procedure logic. Under certain conditions when the first aircraft in the airport-wide queue was unable to depart, other aircraft (in the same or other departure queues at that airport) would not be properly checked for the possibility of a departure.
- 3 Corrected SETCLONE logic. Routes entered by name rather than by number would be ignored, except when included within a wildcard expression. The logic now recognizes both names and numbers.
- 4 Enhanced taxi logic. An optional field SpoolupDlyDist has been added to the AFLINKS inputs. This field defines a distribution in the PROBDIST inputs from which a value is selected to represent the delay that an aircraft will incur to enter this link if the aircraft was required to hold prior to entry.
- 5 Added feature to SIMU26 output. Previously, successive ground hold records of the same type would be written for a given aircraft in the SIMU26 file. If the global variable consolidate_hold_codes is set to 1, successive records will be consolidated into one record whose hold duration field is the sum of the hold durations of consolidated records.
- 6 Enhanced taxi logic. Previously, an aircraft would determine its taxiplan only once. If the new global variable recalculate_taxiplan is set to 1, taxiing aircraft will recalculate an optimal taxiplan from their current ground node each time they finish traversing a ground link.
- 7 Corrected gate assignment logic. In situations where there was no gate available to a departure aircraft, it would receive a tentative assignment to an occupied gate. This tentative assignment would be canceled if another gate became available sooner; however, the cancellation logic would corrupt the capacity data of the tentative gate such that an excessive number of aircraft would be assigned to it later.
- 8 Corrected taxi/de-icing logic. The engine would crash if the de-icing effectiveness time elapsed while an aircraft was on the final link of its taxi movement (typically when arriving at a departure queue).

- 9 Corrected gate logic. In situations where an arrival was assigned to a gate (filling it to capacity) but had not yet reached the gate and a departure was subsequently assigned to the same gate, the departure would begin to wait. However, once the arrival reached the gate and began its debarkation activity, the departure would stop waiting and begin its embarkation activity, violating the gate capacity.
- 10 Corrected runway exit selection logic. Landing roll speeds were incorrectly calculated when an exit was chosen from the RUNWAY_EXITS_LINKS inputs. The aircraft would decelerate to taxi speed at the end of the first runway link. Also, the engine would crash when RUNWAY_EXITS_LINKS were defined while no RUNWAY_EXITS_PROHIBITED_LINKS were defined.
- 11 Corrected runway crossing logic. For cases with a departure queue that is located on the runway, crossing aircraft would ignore the existence of departing aircraft that were released from the queue but had not yet commenced their departure roll.

2.5.1
2002-03-13

- 1 Corrected taxi logic. Aircraft would wait for excessive amounts of time in cases where the aircraft was attempting to enter a DSDPATH yet was not the first aircraft in the DSDPATH entry queue when the DSDPATH became available.
- 2 Enhanced TRACE messages. Added TRACE 247 to provide a message when an aircraft starts and finishes a pushback movement.
- 3 Corrected airspace movement logic. Aircraft separations were not properly imposed in certain situations when a node's separation option was set to anything but 1 or 0101 (denoting that all aircraft should be separated at the node regardless the links used to enter or exit). This would result in an incorrectly lower holding delay at the node for a slower aircraft trailing a faster aircraft.
- 4 Enhanced taxi logic. When set to 1, the existing global variable taxi_on_runway, allows an aircraft to taxi on ground links of an active runway only if the aircraft used that runway for its landing roll. A new global variable taxi_on_all_runways has been implemented such that when set to 1, all aircraft may taxi on any active runway. This variable is zero by default. As before, the logic still selects a taxiplan on an active runway only when no other taxiplans without such links are feasible.

5 Modified taxi logic. Previously, the taxi speed of a runway link was determined using the following priority among the applicable inputs:

- Highest 1. DefRSp of RUNWAYS
- 2. Spd of the applicable AFLINK_SPEED_TYPES record
- 3. TaxiSpd of the applicable AFLINKS record
- Lowest 4. DeftSp of AFLINKS

The new priority is as follows:

- Highest 1. Spd of the applicable AFLINK_SPEED_TYPES record
- 2. DefRSp of RUNWAYS
- 3. TaxiSpd of the applicable AFLINKS record
- Lowest 4. DeftSp of AFLINKS

6 Enhanced towing events. New optional input fields have been added to ARRIVAL and EMPLANE each to provide additional time before or after towing events. For departures, the new field specifies the time that boarding begins which may be later than the end of the towing activity. For arrivals, the new field specifies the time that towing begins which may be later than the end of the unloading activity. Fields have been added for the inputs associated with the turnaround/continuing portion of the ARRIVAL and EMPLANE inputs as well.

7 Corrected taxi logic. When attempting to enter a ground link, an aircraft would not detect that another aircraft that is performing a spool-up delay has already been cleared to enter this link. This would cause a temporary loss of separation.

8 Corrected taxi logic. Taxiing aircraft would not be blocked correctly within the airlink/ground node blocking logic.

9 Corrected taxi logic. In cases where a “towing node” (the node to which an arrival aircraft is towed or the node from which a departure aircraft is towed) is also a gate, ground link blocking logic is activated not for the towing node but for the arrival or departure gate when the aircraft enters or exits the towing node.

2.6.0
2003-06-11

1 Corrected AFLINKBLOCKING logic. In cases where a runway link should block another ground link, blocking would not occur if the blocking aircraft is airborne (SIMU26 TA code). Also, reverse blocking would occur in situations where this was not specified.

- 2 Corrected taxi logic. In cases where a departing aircraft is taxiing from an “off-runway” departure queue to the runway and will take off in the “opposite” direction — other taxiing aircraft will not recognize that such a departure has control of the runway. Consequently, other taxiing aircraft may cross the runway instead of holding for the departing aircraft.
- 3 Corrected runway exit selection logic. High speed exit links were not properly checked for exit prohibitions or taxi constraint limitations.
- 4 Corrected DSDPATH logic. In cases where a ground link is assigned to more than one DSDPATH, it was possible for the engine to enter an infinite loop.
- 5 Corrected GATERWY logic. In cases where an aircraft is tentatively (rather than definitively) assigned to a gate, the GATERWY inputs were ignored.
- 6 Corrected runway exit selection logic. Modifications made to item 3 of version 2.5.1.X introduced a problem with landing rolls ending after the first link of the runway.
- 7 Enhanced DSDPATH logic. Previously, aircraft would be removed from DSDPATHs when arriving at a departure queue or staging/holding pads. Logic has been added to maintain DSDPATH occupancy while the aircraft spends time at these nodes.
- 8 Corrected taxi planning logic. The taxi planning logic was not properly checking for prohibited runway exits for arriving aircraft. The runway exit selection logic was also improved.
- 9 Corrected stagger logic. Previously, a stagger node could only have one set of separations although it could be mated with more than one other stagger node. The engine would permit the stagger inputs to include multiple mates; however, the only the separations associated with the last mate listed would be used.
- 10 Corrected taxi planning logic. In cases where a specific runway exit link was chosen as part of the landing roll calculations, the taxi planning logic would erroneously calculate all possible taxi plans to the destination gate (or staging pad) to have an equal “cost” (likelihood of being chosen).

- 11 Corrected gate assignment logic. In situations where there was no gate available to a departure aircraft, it would receive a tentative assignment to an occupied gate. This tentative assignment would be canceled if another gate became available sooner; however, under certain conditions, the aircraft would not immediately be injected at the newly assigned gate.
- 12 Corrected metering logic. Due to a data indexing error, not all meter post nodes would induce delays at the meter nodes. Under typical conditions, only the first few post nodes defined in `SIMU03` would behave correctly.
- 13 Corrected gate logic. Under certain conditions, departure aircraft tentatively assigned to an occupied gate would continue to wait even after the gate became available.
- 14 Corrected landing logic. A memory fault could occur when an aircraft executed a missed approach using a defined missed approach route.
- 15 Corrected stagger logic. The stagger logic was not using `FinSep` wake turbulence separation values when calculating the estimated time of arrival to an interface node. It only used the `MinSep` values. It also did not consider any arrival spread that may be in affect for the interface node.
- 16 Enhanced landing logic. Previously, the `FinSep` wake turbulence separations and arrival spread factors applied only to the last link of an arrival route. A new global variable `num_final_app_links` allows the user to specify the number of final approach links for which the `FinSep` values and spread factors will apply. If missing, this global value defaults to 1.
- 17 Enhanced departure queue input logic. The engine would crash when reading the `DEPARTURE_Q` inputs when the semicolons are not in very specific positions. This logic can now accommodate a greater range of `DEPARTURE_Q` input formats.
- 18 Corrected towing logic. The engine would not always use the correct `GATETOW` record in situations where a towing node is defined in the `ARRIVAL` or `EMPLANE` event and more than one gate uses the same towing node.
- 19 Corrected towing logic. If a flight record indicates a specific towing area, the logic would ignore any `GATETOW` records that specify a concourse (rather than gate) in combination with that towing area.

- 20 Corrected towing logic. If there are no applicable GATETOW records for a flight, SIMMOD would use the taxipath from GATERWY if an applicable one was available.
- 21 Corrected stagger logic. Aircraft were not correctly estimating the time of arrival at the interface node.
- 22 Corrected dynamic re-routing logic. Aircraft on routes with shared links would switch at the first node in common rather than the shared meter node.
- 23 Corrected dynamic re-routing logic. Aircraft on routes with defined missed-approach paths would not dynamically switch to another route.
- 24 Corrected departure blocking logic. Arrivals taxiing from a runway would not block aircraft from departing gates as specified in DEPBLOCK inputs.
- 25 Corrected arrival spread logic. The arrival spread logic would not implement spread for the percentage (PCT) increase type. The incremental (INC) increase and absolute (ABS) spread types were functioning properly.
- 26 Modification of SIMU26 output. Previously, no code was written for airspace node to ground node transitions (the final airspace node of an arrival route) because no simulation time elapses at this node. Likewise, no record was written for ground node to airspace node transitions (the final node of a takeoff roll). Records for these events are now written.
- 27 Corrected separation logic. Recent changes to the arrival spread logic required changes to separation logic for consistency.
- 28 Corrected SIMU14 output. The value for gate use time (fourth field of the second line) for continuing departures (turnaround flights) did not include the gate turnaround time, only the departure boarding time.
- 29 Corrected SIMU14 output. The values for the total taxi out time (thirteenth field of the first line) and the departure queue delay (twelfth field of the second line) can be incorrect for a departure flight for situations in which the aircraft has left the departure queue and reached the runway yet the runway is fouled.
- 30 Corrected TRACE logic. Erroneous logic caused the engine to fail while writing TRACE messages within the DSDPATH logic.

- 31 Modified DSDPATH logic. The `dsdpath_strict_directionality` global variable was added. When this variable is set to 1, the engine denies entry of a DSDPATH to an aircraft if that aircraft will travel a ground link in a direction opposite to that of another aircraft that is already in the DSDPATH. This is true even if there would be no conflict given the estimated times of arrival at the link.
- 32 Modified departure queue logic. The `dq_hold_until_rwy_clear` global variable was added. When this variable is set to 1, the engine will not release an aircraft from the departure queue unless the runway is clear of other aircraft.
- 33 Corrected procedure input logic. The input logic permitted the first related group to contain up to the number of procedures specified in the `MaxProc` field of the `PROCEDURES` inputs. However, all subsequent related groups were limited to 50 procedures.
- 34 Corrected taxi planning logic. In cases where runway exit link distributions are used, the taxi planning logic would not correctly estimate an aircraft's times of arrival at each node of its taxiplan which could then cause gridlock problems on DSDPATHs.
- 35 Corrected airlink/departure queue blocking logic. In some situations, departures may hold an excessively long time in a departure queue after an airborne aircraft has activated airlink/departure queue blocking.
- 36 Corrected procedure logic. In cases where arrival separations are so small that more than one arrival are performing landing rolls at the same time, the procedure logic may incorrectly impose or fail to impose blocking on related procedures.
- 37 Corrected taxi logic. In some situations, aircraft that had to hold when crossing a runway would not hold for the entire spoolup delay.
- 38 Corrected injection logic. In cases where an aircraft was injected on a meter node using metering strategy 1, the aircraft would hold indefinitely.
- 39 Corrected aircraft group input logic. The input logic was failing to create default holding stacks since none were defined.
- 40 Enhanced flight event logic. The `LoadTime` of the `EMPLANE` event and the `TowTime` of the `ARRIVAL` event no longer require that towing be enabled. A flight will delay at the gate until the `LoadTime` or `TowTime` before resuming its flight actions.

41 Corrected procedure logic. In cases where an arrival route comprises one link and an arrival procedure for that route has a blocking distance greater than the length of the link, arrival procedure blocking would not occur.

2.6.1
2004-01-05

- 1 Corrected RUNWAY_EXITS_PROHIBITED_LINKS logic. A logic flaw would cause an aircraft to ignore a prohibited link.
- 2 Corrected runway crossing logic. In cases where a taxiing aircraft arrives at an active runway and will be able to cross it with no elapsed time at the same time that a departing aircraft has been released from the queue but has not yet reached the same runway, the engine can enter an infinite loop.
- 3 Corrected procedure logic. In cases where an arrival procedure is in a related group with departure procedures, yet none of the arrival/departure procedure distance separations is greater than 0.0, the engine can crash.
- 4 Corrected taxi logic. In cases where a departing aircraft switches departure routes after having been towed, an erroneous message regarding departure queues could occur and crash the engine.
- 5 Corrected RUNWAY_EXITS_PROHIBITED_LINKS logic. Arrival aircraft would ignore a prohibited link that was defined for a runway in the “opposite” direction if at least one prohibited link was defined for the same runway in the “primary” direction.
- 6 Corrected FLOW logic. Internal engine variables related to FLOW were improperly used.
- 7 Corrected takeoff logic. In cases where a taxiing aircraft wants to enter a ground link on an active runway at the same moment another aircraft is performing a takeoff roll on that link, the taxiing aircraft could stop and hold indefinitely.
- 8 Improved airlink/airlink blocking logic. The LINKBLOCKING logic was modified to improve its efficiency and performance.
- 9 Improved RWYCROSS logic. This logic was modified to better identify situations when an aircraft can cross a runway without disturbing the departure and arrival flow.

- 10 Enhanced WINDSET logic. Previously, a maximum of ten wind sets were permitted with a maximum of one hundred airspace links per wind set. These limits are now defaults and may be superseded by the global fields of the WINDSET inputs in SIMU03.
- 11 Improved procedure logic. The max_num_final_arrival_links global variable has been eliminated. Previously, the procedure logic required the specification of the maximum number of links prior to the interface node that an arrival aircraft would begin procedure blocking. Now, the logic automatically determines the appropriate number of links.

6.2.0 2004-05-24	1	Enhanced TAXCHKPT logic. Previously, these inputs had no affect on the simulation. A new field MaxACTresh was added so that if the number of aircraft occupying the “look-ahead” links exceeds this value, a re-plan of the taxiplan is triggered.
	2	Corrected taxi logic. In cases where the engine releases a taxiing aircraft onto the next link of its taxiplan yet the aircraft holds for a spoolup delay, it was possible for another aircraft to subsequently enter the same link, violating the link capacity.
	3	Corrected DSDPATH logic. It was possible for an aircraft to enter a DSDPATH in which there would be a future conflict with another aircraft that was already on the DSDPATH but was holding at the time the entering aircraft performs its checks.
	4	Corrected SIMU10 output. The values for the departure delays did not include the departure queue delay component.
	5	Corrected SIMU04 output. The engine would crash when writing the SIMU04 file at the end of a simulation for scenarios that contain runways with names longer than 19 characters.

6.2.1 2004-06-18	1	Corrected TRACE logic. In some situations, the logic was writing an erroneous message in SIMU04 stating that the procedure blocking distance exceeds the total length of the final arrival links.
---------------------	---	---

6.2.1.1 2004-07-08	1	Corrected runway crossing logic. In situations where a runway crossing was placed at the intersection of two or more runways and the runway crossing gained the priority, departures from those runways could be held indefinitely.
	2	Corrected takeoff logic. Spoolup delay was not applied to an aircraft that was delayed at its procedure start node due to an occupied runway (SIMU26 code WR).

6.2.2 2004-07-23	1	<p>Enhanced DSDPATH logic. Previously, if an aircraft was denied entry to a DSDPATH due to forecasted conflicts with other aircraft already on the DSDPATH, the aircraft would be queued for entry at such time that no movement conflicts would occur. While this first aircraft was waiting, a second aircraft could enter the DSDPATH if there would be no conflicts between this second aircraft and the other aircraft already on the DSDPATH. However, this could cause the first aircraft to wait even longer for the DSDPATH to become clear of conflicts.</p> <p>A new field, MaxWaitThresh, has been added to the DSDPATH inputs of SIMU07. It represents the amount of time (in real seconds) that an aircraft is willing to wait to enter the DSDPATH before enforcing strict first-in-first-out (FIFO) queuing to enter the DSDPATH. If this field is missing, the logic behaves as previously and allows unlimited jumping in the DSDPATH entry queue.</p>
	2	<p>Corrected airlink/departure queue blocking logic. The logic would fail if no departures queues were defined.</p>
	3	<p>Enhanced air logic. Indicated airspeeds between 0 and 10 knots are now permitted.</p>
6.2.3 2004-09-17	1	<p>Corrected airlink/airlink blocking logic. Previously, a blocking airlink could not be entered more than once in the LINKBLOCKING inputs. This limitation is removed.</p>
	2	<p>Enhanced AFLINKBLOCKING input logic. The logic that loaded the AFLINKBLOCKING inputs temporarily consumed large amounts of memory. The new logic stores the inputs more efficiently and loads it such that memory consumption is a monotonic increase.</p>
	3	<p>Corrected GATERWY logic. Previously, the logic would work correctly only for the airport that was listed first in the AIRPORTS input of SIMU03.</p>
6.2.4 2004-11-05	1	<p>Corrected sector capacity logic. A flaw in the logic could cause an aircraft to wait indefinitely for a sector that was unsaturated through a SETSECT.</p>
	2	<p>Improved memory usage. Several logic problems caused excessive memory consumption.</p>
6.2.4.1 2005-01-06	1	<p>Corrected separation logic. The airspace node separation option in the 4-digit format was not applied correctly for aircraft arriving to a node.</p>

- 2 Corrected runway roll logic. The engine would crash if the acceleration/deceleration during the takeoff/landing roll was zero.
- 3 Corrected taxicheckpoint logic. The taxicheckpoint logic would erroneously check for a valid departure queue, even if the aircraft was an arrival.
- 4 Corrected staging logic. Departures would automatically go to a staging pad that was defined for their departure queue/gate/plan combination, even though the maximum queue threshold was not exceeded.

6.2.4.2
2005-03-21

- 1 Corrected runway exit logic. The probabilities specified in the distribution were not properly considered. The logic was not considering directionality properly when choosing exits based on distributions. Distributions defined for either direction would be considered. Also, the logic would always use the value of the displaced threshold in the primary direction for a runway, even when the aircraft was traveling in the opposite direction.
- 2 Corrected SIMU02 output. A number of erroneous messages were eliminated.
- 3 Corrected gate input logic. The gate selection logic was originally designed such that when a flight is choosing from among available gates with identical capacity, preference will be given to those gates that have the fewest airlines assigned to them. This preference was erroneously deactivated in 1999.
- 4 Corrected departure procedure logic. The departure procedure logic was using an arrival tail factor rather than a departure tail factor when calculating the procedure separations for subsequent departures.

7.1
2005-07-05

- 1 Changed node separation logic. The default value of the global variable `node_sep_ac_choice` has been changed from 0 to 2. When 0, the logic used the trailing aircraft's speed in the separation calculations. This approach can result in a loss of separation, whereas using 2, the leading aircraft's speed, will ensure that the desired separation is never lost.
- 2 Corrected aircraft separation logic. When determining the separation of aircraft leaving a node, the logic was erroneously using the trailing aircraft's speed on arrival to the node rather than expected speed upon leaving the node.

- 3 Corrected gate selection logic. The gate selection logic would not function correctly if the airline identifier contained a number.
- 4 Corrected taxi planning logic. If an aircraft was re-assigned to another gate at a taxicheckpoint or staging pad, it would not re-examine the GATERWY inputs to determine if it should use a different taxipath.
- 5 Enhanced runway crossing logic. The multiple_rwcross_delay global variable was added. This variable represents the number of seconds of delay additional applied to each aircraft that are crossing the same runway after a departure or arrival passes. This delay is intended to represent the amount of time required by a ground controller to issue crossing instructions. **FAA**
- 6 Modified procedure logic. The procedure logic allows an aircraft to be released from the departure queue when no other procedures were blocking the departure procedure, even if the runway is occupied. The former global variable DQ_HOLD_UNTIL_RWY_CLEAR value of 1 forced the procedure logic to wait for the runway to be clear before allowing a departing aircraft to be released from a departure queue. This global variable has been replaced by the departure procedure HoldUntilRwyClear input parameter so that departing aircraft using different procedures can behave differently.
- 7 Modified departure queue logic. The departure queue logic allowed passing in the queue if the passing aircraft would not cause the passed aircraft to be delayed (given a later release time). The departure queue PassDelayFlag input parameter has been added so that a value of "P" causes the logic to ignore this rule.
- 8 Modified departure queue logic. The former global variable ALLOW_ALINK_DEPQ_INV_BLK value of 1 enabled any departure queue subject to airlink blocking with the ability to inversely block airlinks. The inverse blocking duration was hard-coded as 30 seconds. This global variable has been replaced by the departure queue InvBlkDur input parameter. A value of 0.0 indicates no inverse blocking. Values greater than zero represent the inverse blocking duration in seconds.
- 9 Corrected pushback logic. In cases where an aircraft blocks ground links during a pushback, The aircraft would not first check that no other aircraft was taxiing on these links prior to commencing the pushback.

- 10 Enhanced departure queue logic. The new input DEPARTQ_GROUP has been created. This input allows the analyst to define groups of departure queues such that all of the member queues of the group are treated as one for the purposes of determining the next departure, queue ordering, and re-routing.
- 11 Modified metering logic. Minor changes were made to improve the forecasting calculations and to ensure that metering vector delay does not exceed the value permitted on a given link. The metering logic allows passing to occur on links that prohibit passing unless the new global variable enforce_metering_no_passing is set to 1.
- 12 Corrected gate logic. In some cases where a flight was blocked from leaving the gate due to an aircraft on a blocking link, the flight would wait an excessive amount of time. Fixed.
- 13 Enhanced landing roll logic. An additional parameter was added to the ExitLnkDist field of the RUNWAY_EXITS_LINKS inputs. This optional parameter allows the analyst to specify a runway occupancy time for a given combination of runway/gate/aircraft model/exit rather than by assuming a specific initial speed over the threshold. The runway roll calculations still assume constant deceleration.
- 14 Enhanced takeoff roll logic. A new input type has been added to the SIMU07 file — RUNWAY_TAKEOFF_OCCUPANCIES. These optional inputs allow the analyst to specify a runway occupancy time for a given combination of runway/aircraft model rather than by assuming a specific speed at the time when the wheels leave the ground. The length of the takeoff roll distance is determined using the same probability value used to choose the occupancy time. The runway roll calculations still assume constant acceleration.
- 15 Corrected gate input logic. The gate input logic would not read all the staging data for the gate in cases where there were more than one plan and the inputs were not grouped by plan.
- 16 Corrected staging logic. The correct staging area would not necessarily be chosen for arrival flights that re-routed.
- 17 Enhanced stagger logic. Staggering would be lost if aircraft held at nodes downstream from the pre-stagger nodes.

The enhancement updates the estimated time of arrival at the runway if downstream holding occurs and imposes additional stagger holding, if necessary, at the pre-stagger nodes. **FAA**

- 18 Enhance SIMU10 output. A new table has been included to provide summary statistics for each departure queue. **FAA**
- 19 Enhanced SIMU14 output. The current format of SIMU14 spreads each record over two lines. A optional global variable simu14_format has been implemented to control the format more precisely as follows:
 - 1 One line per record; include all headers and footers
 - 2 Two lines per record; include all headers and footers (Default)
 - 11 One line per record; include footers but no headers
 - 12 One line per record; include headers but no footers
 - 13 One line per record; no headers or footers
 - 21 Two line per record; include footers but no headers
 - 22 Two line per record; include headers but no footers
 - 23 Two line per record; no headers or footers **FAA**

-
- | | |
|---------------------|---|
| 7.1.1
2005-10-28 | <ul style="list-style-type: none"> 1 Corrected takeoff roll logic. The acceleration was not calculated correctly for flights that used the RUNWAY_TAKEOFF_OCCUPANCIES inputs. 2 Modified SIMU26 output. The SW and SF codes were not being written for departure staging events. 3 Corrected metering logic. Logic changes in the last version to implement the global variable enforce_metering_no_passing resulted in errors that would disable the metering if the value was set to 0. 4 Modified gate holding logic. Previously, if a gate has been designated that aircraft should hold at the gate if the departure queue is full and that aircraft should use departure staging, flights would taxi to the staging area even though the staging area was filled to capacity. Aircraft will now hold at the gate until space is available at the staging area. 5 Modified output logic. The SIMU26 QA code was not correctly being written when the global variable use_depq_hold_codes was set to 1 or 3. Furthermore, departure queue delays that should have been attributed to the QA code were not being accumulated as such in SIMU14. 6 Corrected landing roll logic. Landing aircraft using procedures with an end node at the end of the runway but actually use a high speed exit to complete their roll would not properly block the last node of the runway to other procedures and taxiing aircraft. |
|---------------------|---|
-

-
- 7.2
2006-07-11
- 1 Corrected runway exit selection logic. Exit links were not properly checked if they were available to arrival (versus departure) aircraft.
 - 2 Corrected air logic. Nodes with a control strategy other than “QFIFO” would automatically enforce QFIFO if another aircraft was already holding (due to some reason other than separation requirements) for the node.
 - 3 Corrected stagger logic. In cases where additional stagger holding is applied due to downstream delays, the amount of additional stagger may have been too great.
 - 4 Corrected SIMU14 output. The final field representing the runway crossing delay would not be correct for the departure flight of a turnaround.
 - 5 Corrected gate blocking logic. In some cases where an aircraft must hold at a gate because the gate-blocking links are occupied, the aircraft may not push back from the gate at the earliest possible opportunity but instead incur excessive delay with a PD code in the SIMU26 output.
 - 6 Corrected holding stack logic. Previously, the engine would break if more than ten holding stack were defined. Now the number of stacks is limited to the number of nodes.
 - 7 Changed node separation logic. The default value of the global variable `node_sep_ac_choice` has been changed from 2 to 1. This was incorrectly changed from 0 to 2 in version 7.1.
 - 8 Corrected touch-and-go logic. The logic related to the PATTERN and TNGSET inputs was re-implemented to eliminate extraneous fields in the PATTERN inputs.
 - 9 Recompiled with latest compiler. Internal array dimensions are no longer limited to 65536.
 - 10 Modified AN_MDL_GATE_CHOICE inputs. Previously, the MdlList was not allowed to be missing. This list is now allowed to be missing as it is interpreted to mean that all aircraft models match the input.
 - 11 Enhanced to AFLINKS inputs. New optional restrictions can be specified to prevent usage of the ground link: AIRLINE to restrict airlines, ARLN_MODEL to restrict airline/aircraft model combinations, TIME to restrict taxiplanning during specified periods. **FAA**

- 12 Added HOLDING_APRON inputs. Defines nodes at which aircraft temporarily hold. **FAA**
- 13 Enhanced DEPARTQ_GROUPS. Added Thresh and WaitThresh fields to allow departure queue groups to trigger arrival spread.
- 14 Added metering logic options. Two new versions of the metering logic have been implemented. To activate these versions, specify a GLOBAL data input "new_metering_logic" with a value of "1" or "2". Value "2" is more robust in which aircraft do not hold between the meter and post nodes. If this GLOBAL input is missing or equal to "0", the engine uses the legacy metering logic.

7.2.1
2006-08-23

- 1 Corrected stagger logic. Eliminated round-off errors in delay calculations that could result in an infinite loop.
- 2 Corrected taxi logic. When reaching a gate or tow node, aircraft now continue to occupy the link on which they used to enter the node until they either taxi away from the node or until they leave the simulation.
- 3 Corrected towing logic. In cases where a tow node is also a gate node, any aircraft towed to the node will also occupy one unit of space at the gate.

7.2.2
2006-12-15

- 1 Corrected taxiplanning logic. The estimated times of arrival at each node of a taxiplan were not correct for towing aircraft if the towing speed was different than the taxi speed.
- 2 Enhanced towing logic. If the gate of a towed departure is occupied, the departure will wait at the tow node until that (or another feasible) gate becomes available before performing the towing movement.
- 3 Corrected SIMU14 output. In some cases, the gate use and delay fields were incorrect for flights that were towed.
- 4 Corrected stagger logic. Internal variables associated with the stagger logic were not re-initialized with each iteration or when an aircraft left the simulation. Fixed.
- 5 Corrected AFLINKBLOCKING input logic. A flaw was fixed in the portion of the AFLINKBLOCKING input logic that automatically discards duplicate combinations.
- 6 Corrected runway exit logic. The inputs were read correctly in some cases where they were defined for aircraft models.

- 7 Modified gate selection logic. By default, when choosing a gate, a flight will prefer gates that used by a fewer number of airlines. This bias can be disabled by setting the new global variable `disable_gate_airline_bias` to 1.

7.3
2007-09-10

- 1 Enhanced runway exit logic. If no gates are specified in an entry in the `RUNWAY_EXITS_LINKS` and `RUNWAY_EXITS_PROHIBITED_LINKS` inputs, the entry will apply to all gates for that entry.
- 2 Corrected arrival re-routeing. It was possible for a flight to re-route even though the new route did not pass through the same meter post node. Fixed and trace 70 will provide a warning.
- 3 Modified taxi logic. Conflicts between the `DSDPath` logic and the groundlink entry checks could cause gridlocks in rare cases. Fixed.
- 4 Modified departure logic. Aircraft that were blocked (`SIMU26 PD` code) from pushing back due to occupied ground links would not necessarily begin to move immediately after the blocking ended. Fixed.
- 5 Corrected taxi planning logic. If a towed departing aircraft rolls over a taxi checkpoint, the resulting taxiplan will not properly cause the aircraft to be towed to its gate. Fixed.
- 6 Corrected `SETGATE` logic. In cases where the optional `ADistNo` or `DDistNo` is missing, the following event in the `SIMU09` would not be read properly.
- 7 Corrected runway crossing logic. Taxiing aircraft whose taxiplan crosses both end nodes of a runway crossing would automatically be subjected to delays imposed on that runway crossing even if the taxiplan does not actually use that crossing. Fixed.
- 8 Corrected gate holding logic. In cases where departures wait at gates due to departure queue congestion, the departures would not necessarily leave for the departure queue in the order in which they began waiting. Now, such departures pushback in the order in which they began waiting.
- 9 Enhanced runway inputs. Two new fields have been added to the runway inputs — `TaxiFlagP` and `TaxiFlagO` — to indicate if taxiing is allowed on the runway when it is active. These replace the global dat entries of `taxi_on_runway` and `taxi_on_all_runways`.

- 10 Enhanced the SETRWY event. A new entry has been added to the SETRWY event, that allows the analyst to change the taxi flag.
- 11 Enhanced the SETGATE event. A new entry has been added to the SETGATE event, that allows the analyst to change the gate capacity.
- 12 Corrected runway crossing logic. Internal variables associated with the runway crossing logic were not re-initialized with each iteration. This could cause departures in subsequent iterations to hold indefinitely. Fixed.
- 13 Corrected taxi planning logic. Taxipaths were not properly selected in cases where a plan change is in effect and GATERWY inputs are also defined. Fixed.

Last updated: 2007-09-10