



SATURN TEN – Version 10.9.17

First of all, I wish you a warm welcome to the second full release of SATURN 10.9, 10.9.17.

SATURN 10.9 is the successor to all previous versions of SATURN. It represents a direct evolution from its predecessors whilst offering significant enhancements from the strong existing base. The first internal release of SATURN 10.9 was in December 2008 (denoted by purely temporary version numbers 10.9.01 through to 10.9.11) with the first full release 10.9.12 in November 2009, albeit with certain new (non-default) options effectively still in Beta form.

Version 10.9.17 has been issued in June 2010 in order to correct a number of bugs which had been found in 10.9.12 (some minor, some more major; see Appendix E.7 for a full list) as well as to introduce a number of new features as described in Appendix D.18 with highlights in part 3 below. Note that, as with 10.9.12, a number of the more recent features are still effectively in **Beta release** whereas most of the Beta options in 10.9.12 have now been sufficiently tested in practice as to make them “standard” features.

1. Which Version to Use When?

We do not necessarily recommend that users adopt **all** the latest features of 10.9.17; a “mix and match” approach with some programs from 10.9.17 coupled with others from 10.9.12 (or even v10.8.22 if still used) is probably a wiser and safer strategy.

In very broad terms **SATURN** programs may be sub-divided into number crunching assignment plus simulation (**SATNET** plus **SATALL**) and all the others (network analysis, matrix manipulation, etc., etc.). Our advice is that **all** the analysis programs from 10.9.17 may be used straight away but users may wish to continue to use earlier versions of **SATNET/SATALL** in the short term. And at the same time caution needs to be exercised when using Beta features in both earlier releases and 10.9.17 (see part 4 below for more details).

Thus analysis/manipulation programs such as **P1X**, **MX**, **SATPIJA**, **SATME2** etc. may all be used with .ufs and .ufm files produced by earlier releases (within reason; see part 6 below for a discussion on compatibility), various bugs from 10.9.12 will have been corrected and new features added in 10.9.17. We therefore recommend them for immediate use (although clearly there is always a danger of new bugs having been introduced between 10.9.12 and 10.9.17).

In terms of **SATALL** there have been relatively few new features introduced between 10.9.12 and 10.9.17 (see part 3 below) but users will no doubt wish to err on the side of caution by continuing to use either 10.9.12 or 10.8.22 for existing studies.

On the other hand we would strongly recommend that users run their networks through 10.9.17 **SATNET** since it contains a certain number of totally new checks for coding errors plus a large number of new Serious Warnings and Non-Fatal Errors which have been upgraded to Semi-Fatal under **WRIGHT = T**. Indeed most networks that would have “passed” under 10.9.12 will probably “fail” under 10.9.17. Note as well the most recent options in **P1X** for highlighting and correcting network errors. While, for various reasons, you may not wish to change existing networks to eliminate errors which are currently judged to be **FATAL** you should at least be aware of them. These errors may be downgraded back to their original status by using the **SATNET** **ERRYES** option (see section 6.12.2 of the latest manual)

2. *New General Features with SATURN v10.9*

SATURN 10.9 adds many new features affecting simulation-assignment functions and network displays and editing. This section primarily highlights those that were available in 10.9.12 whereas the next section describes the most recent additions in 10.9.17.

- ◆ Various updates to the **SATWIN** Interface with the option to link the various files used by SATURN to their respective programs (ie executables) with Windows Explorer (see 3.6.6) as well as re-running previous events (see 3.6.5)
- ◆ A new command line parameter **QUICK** in the key programs, if used, artificially reduces all loop etc. counters such as MASL, NITA, NITS etc. to minimal values so that the program runs in minimal CPU. This is very useful for checking if long batch files have been set up correctly (see 14.10). The QUICK option may also be set globally (along with the existing **QUIET** option) via **SATWIN** (see 15.55)
- ◆ A significant number of further 'error' checks in **SATNET**, coupled with the extension of the **WRIGHT** parameter that converts certain existing warnings, serious warnings and/or non-fatal errors into semi-fatal errors (NAFF) on the basis that there is no conceivable reason that could explain these errors. The removal of these coding errors will improve the convergence in the assignment and reduce CPU time
- ◆ A new parameter **FLAREX** has been added in **SATNET** to model explicit flares for X-turns at signals (see 8.2.5.2). A similar option for priority junctions is under development. In conjunction with these changes, **MONACO** and **AUTNUC** now default to **TRUE**
- ◆ There are two significant revisions to the modelling of blocking back within **SATALL** namely:
 - ◆ The network coding practice of sub-dividing a link into a short series of two-arm "artificial" nodes (e.g., to give it "shape") has now been formally identified as a "chain" of nodes with applications to blocking back, random delays, etc. etc. where we expect the series of links to function as a single link in some respects. This will improve the performance of SATURN, the convergence and the presentation of the model outputs (see 5.1.12 and 8.5.5)
 - ◆ Blocking back may also now be "phased in" when the queue on the link is only slightly smaller than the stacking capacity by setting $BB109 = T$ and $BBKING < 1.0$ (see 8.5.6). This is released in **BETA** form so it should be used for **research** purposes and we will be interested in feedback
- ◆ A new option **UFC109** in **SATALL**, if used, will: (a) store the multiple user class link times per iteration as opposed to link (generalized) costs by both iteration and user class and (b) store the costs/times as a rolling summation of **all** Frank-Wolfe iterations over **all** simulation-assignment loops instead of re-creating the assigned route flows by an extra **SAVEIT** assignment. Point (a) means that the **MUC .UFC** files produced under **UFC109** are reduced in size by a factor of $1/NOMADS$ while (b) guarantees that any secondary analysis, e.g., skimming, based on routes is exact, not an approximation. However, there may be a dis-benefit under (b) in that there may be many more rolling iterations in total than there would be in a **SAVEIT** assignment which means that: (i) the **.UFC** files become larger and ii) any secondary analyses take longer (see 15.23.3)
- ◆ The method for calculating random delays upstream of links which block back has been modified so that the capacity used in the formula is the capacity **before** blocking back was applied. This removes a possible discontinuity when the link

downstream goes from a state of blocking back to not blocking back or vice versa. See 8.6.5. In addition random delays are **not** included on links which are part of “internal” chains of 2-arm “artificial” nodes. See 8.6.4. Taken together these may produce significant differences in delay calculations (but clearly only if LRTP > 0)

- ◆ Within **P1X** the differences in network coding of individual simulation nodes between networks 1 and 2 may now be “highlighted” and node graphics loops over those nodes that differ may be initiated. This is a very useful option for discovering the variations between two networks. A complete table of differences per node is printed out automatically to the .LPP file (see 11.6.5.4)
- ◆ An option (SKIM_ALL) to skim O-D time, distance and tolls simultaneously from a forest (as opposed to carrying out 2 or 3 separate skims in order to reduce CPU time by a factor of roughly 2 or 3) has been added in **SATLOOK** (see 15.27.7 and previously introduced with v10.8.21)
- ◆ A new program called **CASSINI** has been added to enable the network parameters to be adjusted between external demand loops dependent on the level of convergence reported by the demand model (eg DIADEM). This may significantly reduce the SATURN assignment times for the earlier loops depending on the settings used (see 15.54 and Appendix R)

We also mention that, from v10.8.17 onwards, a new upper limit of **2000** zones in the *standard* version of MX was introduced to reduce excessive memory requirements; uncapped versions permitted within the licence are freely available upon request without additional charge. Whilst the overall limit remains, the internal limits have been increased to provide more ‘headroom’ for stacked matrices. If users receive an error message stating that there are too many rows, please contact us and we will supply the larger version free of charge.

3. Latest Additions in 10.9.17

This section reviews the most important changes which have been made between 10.9.12 and 10.9.17 (i.e., between November 2009 and June 2010). References to the relevant sections in the Manual are given.

- ◆ *Network Aggregation (Spider Web Networks) (15.56)*. A very simple but very powerful idea it works by combining links in series prior to carrying out a network assignment and empirically reduces assignment CPU times by factors of between 4 and 10 or, in certain cases, even more. It can equally reduce CPU times for post-assignment analyses such as cost skims. Note that it does not reduce simulation CPU time by similar amounts so that the overall reductions in CPU time are somewhat less – but still comparable to using multi-core. N.B. Still in Beta form.
- ◆ *Tighter SATNET error checking with WRIGHT = T (6.12.2)*. A significant additional number of Serious Warnings and Non-Fatal Errors have been upgraded to Semi-Fatal Errors under WRIGHT = T in order to better reflect what we feel current practice should be. Most previously coded networks are liable to fail under the latest criteria although by a judicious use of ERRYES exemptions and/or setting WRIGHT = F it is quite feasible to continue to run old networks. But not necessarily recommended!
- ◆ *Changes to TOPUP in SATNET (6.15.3)* which make it easier to over-write simulation node coding with \$INCLUDE files and within the same file by setting DOUBLE=T.

- ◆ *An accelerated version of AUTOK (9.3.2)* may reduce the number of repeated simulations necessary to average assignments and therefore reduce overall CPU time.
- ◆ *A new “topological” order of node simulation (8.3.4)* has been introduced with a view to speeding up simulation.
- ◆ *Extra link etc. data items* added to standard lists in P1X
- ◆ *A “Resume” option in P1X (11.4.4)* allows users to carry on using P1X with exactly the same set of parameters and window definitions as at the conclusion of the previous run of P1X.
- ◆ *Error Report Log (ERL) files (15.58)* are an attempt to make it easier to identify coding errors which have previously been checked (therefore part of an “audit trail”) as well as to distinguish newly added errors. Still very much under development.

4. *Beta Release Options etc.*

We list here those features of 10.9.17 (and/or 10.9.12) which should still be regarded as Beta releases and under development rather than fully validated (although the degree of validation varies *widely between those listed* below as indicated by Beta plus/minus etc.):

- ◆ *BB109 and BBKING (See 8.5.6)*. First introduced in 10.9.12 we are not aware of any problems and they do seem to help convergence but they have not been used all that much that we know of. Beta minus.
- ◆ *UFC109 (See 15.23.3)* Introduced in 10.9.12 it caused problems, particularly in conjunction with SATPIJA, but hopefully all problems have been corrected under 10.9.17. Beta minus.
- ◆ *FLAREX/FLAREF (See 8.2.5.2)* Introduced in 10.9.12, the original function of FLAREX to allow extra capacity for straight ahead traffic in a shared lane with blocked X-turners at signals appears to be stable (i.e., no problems have been reported) but it has been extended to model priority junctions and inner lanes (FLAREF) as well. Beta plus for the latter function.
- ◆ *Network Aggregation (See 15.56)* Introduced in 10.9.17 we are so far unaware of any problems but it is early days yet and full analysis options are yet to be finalised. Beta.
- ◆ *Error Report Log (ERL) files (See 15.58)*. A good idea still in its infancy: Beta plus.

5. *SATURN Algorithm: Origin-Based Assignment for Multiple User Classes*

The Multiple-User Class version of the Origin-based assignment algorithm (OBA-MUC) was released as a **standard** part of the SATURN suite with v10.9.12 to all our users and is therefore included within 10.9.17 as well.

The testing work on OBA-MUC algorithm, as reported at the 2009 European Transport Conference (see Appendix S), has demonstrated that, for the majority of networks, OBA-MUC is able to achieve a higher level of convergence than the existing FW algorithm for the same CPU expenditure.

These higher levels of convergence now achievable in SATURN will provide practical benefits to other models that are sensitive to convergence including demand models (e.g. DIADEM) and cost-benefits models (e.g. TUBA) for example. In addition, as the



algorithm stores the route proportions (in the .UFO file), any secondary analysis may be undertaken without needing to re-build the paths and thereby considerably reducing the CPU required. A follow-up paper will be presented at the 2010 ETC and the abstract is reproduced in Appendix S.

The current OBA-MUC development work has also reinforced the importance of good quality network coding (as enforced by WRIGHT=T). Convergence within SATURN is sensitive to the interactions between the assignment and simulation loops. The new OBA algorithm is able to achieve very high levels of convergence but only if the network coding enables robust and stable estimates of flow and travel costs to be calculated.

6. *Compatibility with Previous Versions*

In terms of compatibility with previous releases, there are a number of points that users need to be aware of:

- ◆ There have been no changes to the format of binary matrix (.ufm) files so that matrix files from previous releases (within reason!) may still be used within 10.9.17 and, conversely, previous releases of programs such as MX should be able to happily read and process .ufm files produced under 10.9.17;
- ◆ The formats of network ASCII data files (.dat) and (a few) control files have evolved marginally (e.g., new &PARAM parameters and/or options have been added) such that there is no absolute guarantee that a file created under 10.9.17 “rules” will be backwards compatible for programs issued under previous releases. However 10.9.17 programs should be able to correctly read all data and binary .ufs files created by previous SATURN releases;
- ◆ There is, however, a caveat to the last statement given that the number of error checks, both fatal and non-fatal, has increased in 10.9.17 (and in 10.9.12 as well) particularly with the WRIGHT parameter. It is therefore **probable** that network data files which previously “passed” will now “fail” in SATNET under one of the new fatal errors and the necessary corrections will need to be made before that network will run under 10.9. Alternatively, the user may turn this ‘off’ by setting WRIGHT=F;
- ◆ Given the changes introduced into some of the simulation and assignment routines (e.g., to prevent program crashes, hard-coding Q106 and RB106 to TRUE and setting the defaults for RTP108, MONACO, AUTONA, AUTOK and AUTNUC to TRUE) SATURN 10.9.12 will almost certainly produce different answers, given the same inputs, from previous releases, say, 10.8.22 and 10.9.17 will equally give different results to 10.9.12. If users wish to try out the phased-in Blocking Back (BBKING) option then the differences are likely to be much larger. To a certain extent older results may be reproduced by using the parameter NFT = 10.8 (say) to exclude the most recent changes but, unfortunately, some of the changes have had to be “hard wired” into the code and cannot be avoided. Remember that they have all been introduced for very good reasons!

7. *SATURN Multi-Core*

One of our other development activities over the last 12 months was the release of the commercial add-on **SATURN Multi-Core**, a multi-threaded version of the existing SATALL that is able to take full advantage of the additional processors (or cores) available on most Intel / AMD-powered standard desktop PCs. The recent testing across a range of different sized SATURN models has demonstrated the significant



reductions in model runtimes. Typically, SATURN Multi-core reduced the overall model runtimes by up to 1 / N where N is the number of physical cores available depending on the size and type of network and the assignment parameters used.

We are completing the development of a distributed version of SATPIJA which will be available in early July with a multi-threaded version SATLOOK to follow. These will be released, free of charge, to the users that have purchased the multi-core add-on module.

8. Future Releases of 10.9 and Beyond

It is intended that one more full version of 10.9, i.e., beyond 10.9.17, will possibly be released within, say, the next three months in order to complete various recent extensions but also, inevitably, to correct whatever problems may arise with those new components of 10.9.17. In particular we envisage further testing of and modifications to the network aggregation routines, extensions to multi-core (specifically SATPIJA, SATLOOK and Network Aggregation) and more work on .ERL files. We anticipate that the first two will lead to significant reductions in CPU times such that, when these are in a sufficiently stable state, we envisage moving to an initial beta release of **SATURN 11.1**, provisionally scheduled for autumn 2010.

In the meantime, and whether or not there is a full release of 10.9.17+ or whether it is overtaken by 11.1, we propose to issue updated versions of **P1X** and **MX**, i.e., **P1XB** and **MXB**, at appropriate intervals since they do not need to be part of a full release.

9. Compatibility with Windows 7

SATURN v10.9 is fully compatible with the latest release of Windows 7. Whilst SATURN continues as a 32-bit application, it will run equally well in either the 32-bit or 64-bit versions.

10. Technical Support

During the last six months, we have now migrated our all enquires and support over to our new Customer Relationship Management System with e-mails sent through our common e-mail address – saturnsoftware@atkinsglobal.com. Our existing individual e-mail addresses will continue to be available but the common e-mail address will enable us to improve communications with our users.

If you have any suggestions for what you would like to see in future versions of SATURN and SATWIN, please let us know. We rely on you, as users, to move forward. How can you help? Let us know – preferably via the Software Forum (www.saturnsoftware.co.uk/forum) - of any problems you have or features you would like and we will look into them.

In the meantime, I wish you a successful continuation with SATURN 10.9 and thank you for your support over the last 12 months.

Ian Wright

SATURN Product Manager
Transport Planning & Management

ATKINS

ian.wright@atkinsglobal.com

22/06/10 *ReadmeTen9.DOC*

Dirck Van Vliet

SATURN Developer

dirck_van_vliet@yahoo.co.uk